

Liquid War 6

A unique multiplayer wargame
Documentation for version 0.0.9beta codename “Chuck” stamp 2096
23 August 2010

by Christian Mauduit <ufoot@ufoot.org>

Liquid War 6, a unique multiplayer wargame.

Copyright (c) 2005, 2006, 2007, 2008, 2009, 2010 Christian Mauduit <ufoot@ufoot.org>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Table of Contents

1	Introduction	1
1.1	In a nutshell	1
1.2	Project status	1
1.2.1	What works, and what does not (yet)	1
1.2.2	What has changed since Liquid War 5.x?	2
1.2.3	Revision history	2
1.2.4	Road map	3
1.3	How you can help	3
1.3.1	Help GNU	3
1.3.2	Todo list	3
2	User's manual	5
2.1	Mailing lists	5
2.1.1	General discussion	5
2.1.2	Announcements	5
2.1.3	Bugs	5
2.2	Getting the game	5
2.2.1	Download source	5
2.2.2	Download binaries	6
2.2.3	GIT repository	6
2.2.4	Daily snapshots	6
2.3	Installation	6
2.3.1	Requirements	6
2.3.2	Optional libraries	7
2.3.3	Installing requirements using RPM/DEB packages	8
2.3.4	Compiling	8
2.4	Extra maps	9
2.4.1	The extra maps package	9
2.4.2	Install extra maps on GNU/Linux and POSIX systems	9
2.4.3	Raw install of extra maps (all-platforms)	9
2.5	Troubleshooting	9
2.5.1	Compilation problems	10
2.5.2	Check installation	10
2.5.3	Problems running the game	10
2.6	Quick start	11
2.6.1	Quick start	11
2.7	Strategy tips	11
2.8	User interface	11
2.8.1	A reduced set of keys	11
2.8.2	Combining mouse, keyboard and joysticks	12
2.8.3	Quit with F10	12
2.9	Network games	12
2.10	Graphics	12

2.10.1	Standard, high and low resolution	12
2.10.2	Display rate	13
2.11	Sound & music	13
2.11.1	Current status	13
2.11.2	The future	13
2.12	Config file	13
2.13	Logs	14
2.14	Report bugs	14
3	Hacker's guide	15
3.1	Designing levels	15
3.1.1	Why is level design so important?	15
3.1.2	Format overview	15
3.1.3	map.png	15
3.1.4	layer2.png ... layer7.png	16
3.1.5	texture.png, texture.jpeg and texture-alpha.jpeg	16
3.1.6	glue.png and boost.png	17
3.1.7	danger.png and medicine.png	17
3.1.8	one-way-<direction>.png	18
3.1.9	rules.xml	18
3.1.10	hints.xml	19
3.1.11	style.xml	19
3.1.12	Resampling	19
3.1.13	Music	20
3.2	Translating	20
3.2.1	Using gettext	20
3.2.2	Formatted strings	20
3.2.3	Partial translation	21
3.3	Architecture	21
3.3.1	C + Guile	21
3.3.2	Threading and SMP	22
3.3.3	Internal libraries	23
3.4	Memory structures	24
3.5	100% predictable algorithm	25
3.6	About mod-gl	25
3.6.1	The main renderer	25
3.6.2	Hardware requirements	26
3.6.3	The gl-utils toolbox	26
3.7	Compilation tips	26
3.7.1	Advanced ./configure options	26
3.7.2	Microsoft Windows msys/mingw32 port	27
3.7.3	Mac OS X port	31
3.7.4	GP2X port	33
3.8	Coding guidelines	34
3.8.1	Project goals reminder	34
3.8.2	Common sense	35
3.8.3	Unitary tests	35
3.8.4	Memory allocation	35

3.8.5	Private and public interfaces	36
3.8.6	Audit the code	36
3.9	Using the console	37
3.10	Advanced tweaking	37
3.10.1	Hacking ressources	37
3.10.2	Optimize for speed	38
3.11	Writing modules	38
3.12	Use as a library	38
3.13	Network protocol	38
3.13.1	No server, no client, only nodes	38
3.13.2	Out of band messages	39
3.13.3	Other raw technical stuff (WIP)	39
3.14	Using GNU Arch	40
3.14.1	About GNU Arch	40
3.14.2	Getting the latest version from the repository	41
3.14.3	Setting up your own arch repository	41
3.14.4	Synchronizing your repository with upstream releases	42
3.14.5	Submitting patches	42
3.14.6	Recover from broken lock	42
3.15	Using GIT	42
3.15.1	About GIT	42
3.15.2	Getting the latest source	43
3.15.3	Developer access	43
3.15.4	Submitting patches	43
4	Reference	45
4.1	Basic options	45
4.1.1	about	45
4.1.2	audit	45
4.1.3	copyright	45
4.1.4	debug	45
4.1.5	defaults	45
4.1.6	help	46
4.1.7	host	46
4.1.8	list	46
4.1.9	modules	46
4.1.10	pedigree	46
4.1.11	test	46
4.1.12	version	46
4.2	Doc options	46
4.2.1	example-hints-xml	46
4.2.2	example-rules-xml	47
4.2.3	example-style-xml	47
4.2.4	list-aliases	47
4.2.5	list-doc	47
4.2.6	list-funcs	47
4.2.7	list-graphics	47
4.2.8	list-hooks	47

4.2.9	list-input	47
4.2.10	list-map	47
4.2.11	list-map-hints	48
4.2.12	list-map-rules	48
4.2.13	list-map-style	48
4.2.14	list-network	48
4.2.15	list-path	48
4.2.16	list-players	48
4.2.17	list-quick	48
4.2.18	list-show	48
4.2.19	list-sound	48
4.2.20	list-tuning	49
4.3	Show options	49
4.3.1	show-build-bin-id	49
4.3.2	show-build-cflags	49
4.3.3	show-build-codename	49
4.3.4	show-build-configure-args	49
4.3.5	show-build-copyright	49
4.3.6	show-build-datadir	49
4.3.7	show-build-date	49
4.3.8	show-build-docdir	50
4.3.9	show-build-enable-allinone	50
4.3.10	show-build-enable-console	50
4.3.11	show-build-enable-fullstatic	50
4.3.12	show-build-enable-gcov	50
4.3.13	show-build-enable-gprof	50
4.3.14	show-build-enable-gtk	50
4.3.15	show-build-enable-instrument	50
4.3.16	show-build-enable-mod-csound	51
4.3.17	show-build-enable-mod-gl	51
4.3.18	show-build-enable-mod-http	51
4.3.19	show-build-enable-mod-ogg	51
4.3.20	show-build-enable-optimize	51
4.3.21	show-build-enable-paranoid	51
4.3.22	show-build-enable-profiler	51
4.3.23	show-build-enable-valgrind	52
4.3.24	show-build-endianness	52
4.3.25	show-build-gcc-version	52
4.3.26	show-build-gp2x	52
4.3.27	show-build-host-cpu	52
4.3.28	show-build-host-os	52
4.3.29	show-build-hostname	52
4.3.30	show-build-includedir	52
4.3.31	show-build-ldflags	52
4.3.32	show-build-libdir	53
4.3.33	show-build-license	53
4.3.34	show-build-localedir	53
4.3.35	show-build-mac-os-x	53

4.3.36	show-build-md5sum	53
4.3.37	show-build-ms-windows	53
4.3.38	show-build-package-name	53
4.3.39	show-build-package-string	53
4.3.40	show-build-package-tarname	53
4.3.41	show-build-pointer-size	54
4.3.42	show-build-prefix	54
4.3.43	show-build-stamp	54
4.3.44	show-build-time	54
4.3.45	show-build-top-srcdir	54
4.3.46	show-build-url	54
4.3.47	show-build-version	54
4.3.48	show-build-x86	54
4.3.49	show-config-file	54
4.3.50	show-cwd	55
4.3.51	show-data-dir	55
4.3.52	show-default-config-file	55
4.3.53	show-default-data-dir	55
4.3.54	show-default-log-file	55
4.3.55	show-default-map-dir	55
4.3.56	show-default-map-path	55
4.3.57	show-default-mod-dir	55
4.3.58	show-default-music-dir	56
4.3.59	show-default-music-path	56
4.3.60	show-default-prefix	56
4.3.61	show-default-script-file	56
4.3.62	show-default-user-dir	56
4.3.63	show-log-file	56
4.3.64	show-map-dir	56
4.3.65	show-map-path	56
4.3.66	show-mod-dir	57
4.3.67	show-music-dir	57
4.3.68	show-music-path	57
4.3.69	show-prefix	57
4.3.70	show-run-dir	57
4.3.71	show-script-file	57
4.3.72	show-user-dir	57
4.4	Path options	57
4.4.1	config-file	58
4.4.2	data-dir	58
4.4.3	log-file	58
4.4.4	map-dir	58
4.4.5	map-path	58
4.4.6	mod-dir	59
4.4.7	music-dir	59
4.4.8	music-path	59
4.4.9	prefix	60
4.4.10	script-file	60

4.4.11	user-dir	60
4.5	Graphics options	61
4.5.1	capture	61
4.5.2	fullscreen	61
4.5.3	gfx-backend	61
4.5.4	gfx-quality	61
4.5.5	height	62
4.5.6	width	62
4.5.7	windowed-mode-limit	62
4.6	Sound options	62
4.6.1	ambiance-exclude	62
4.6.2	ambiance-file	63
4.6.3	ambiance-filter	63
4.6.4	music-volume	63
4.6.5	snd-backend	63
4.6.6	sound-volume	63
4.7	Network options	64
4.7.1	bind-ip	64
4.7.2	bind-port	64
4.7.3	cli-backends	64
4.7.4	known-nodes	64
4.7.5	node-description	65
4.7.6	node-title	65
4.7.7	password	65
4.7.8	public-url	65
4.7.9	skip-network	66
4.7.10	srv-backends	66
4.8	Map parameters	66
4.8.1	chosen-map	66
4.8.2	force	66
4.8.3	use-hints-xml	67
4.8.4	use-music-file	67
4.8.5	use-rules-xml	67
4.8.6	use-style-xml	67
4.8.7	use-texture	68
4.9	Map rules.xml	68
4.9.1	boost-power	68
4.9.2	color-conflict-mode	68
4.9.3	cursor-pot-init	68
4.9.4	danger-power	69
4.9.5	fighter-attack	69
4.9.6	fighter-defense	69
4.9.7	fighter-new-health	69
4.9.8	fighter-regenerate	70
4.9.9	frags-fade-out	70
4.9.10	frags-mode	70
4.9.11	frags-to-distribute	71
4.9.12	glue-power	71

4.9.13	max-cursor-pot	71
4.9.14	max-cursor-pot-offset	71
4.9.15	max-nb-cursors	72
4.9.16	max-nb-nodes	72
4.9.17	max-nb-teams	72
4.9.18	max-round-delta	72
4.9.19	max-zone-size	72
4.9.20	medicine-power	73
4.9.21	moves-per-round	73
4.9.22	nb-attack-tries	73
4.9.23	nb-defense-tries	74
4.9.24	nb-move-tries	74
4.9.25	respawn-position-mode	74
4.9.26	respawn-team	74
4.9.27	round-delta	75
4.9.28	rounds-per-sec	75
4.9.29	side-attack-factor	75
4.9.30	side-defense-factor	76
4.9.31	single-army-size	76
4.9.32	spread-thread	76
4.9.33	spreads-per-round	76
4.9.34	start-blue-x	77
4.9.35	start-blue-y	77
4.9.36	start-cyan-x	77
4.9.37	start-cyan-y	77
4.9.38	start-green-x	78
4.9.39	start-green-y	78
4.9.40	start-lightblue-x	78
4.9.41	start-lightblue-y	78
4.9.42	start-magenta-x	78
4.9.43	start-magenta-y	79
4.9.44	start-orange-x	79
4.9.45	start-orange-y	79
4.9.46	start-pink-x	79
4.9.47	start-pink-y	80
4.9.48	start-position-mode	80
4.9.49	start-purple-x	80
4.9.50	start-purple-y	80
4.9.51	start-red-x	80
4.9.52	start-red-y	81
4.9.53	start-yellow-x	81
4.9.54	start-yellow-y	81
4.9.55	total-armies-size	81
4.9.56	total-time	82
4.9.57	vertical-move	82
4.9.58	x-polarity	82
4.9.59	y-polarity	83
4.9.60	z-polarity	83

4.10	Map hints.xml	83
4.10.1	background-color-auto	83
4.10.2	downsize-using-bench-value	83
4.10.3	downsize-using-fighter-scale	84
4.10.4	fighter-scale	84
4.10.5	guess-colors	84
4.10.6	guess-moves-per-sec	85
4.10.7	hud-color-auto	85
4.10.8	max-map-height	85
4.10.9	max-map-surface	85
4.10.10	max-map-width	86
4.10.11	menu-color-auto	86
4.10.12	min-map-height	86
4.10.13	min-map-surface	87
4.10.14	min-map-width	87
4.10.15	resample	87
4.10.16	speed	87
4.10.17	system-color-auto	88
4.10.18	upscale-using-bench-value	88
4.10.19	upscale-using-fighter-scale	88
4.10.20	view-color-auto	88
4.10.21	wall-grease	89
4.11	Map style.xml	89
4.11.1	animation-density	89
4.11.2	animation-speed	89
4.11.3	background-color-root-bg	90
4.11.4	background-color-root-fg	90
4.11.5	background-color-stuff-bg	90
4.11.6	background-color-stuff-fg	91
4.11.7	background-style	91
4.11.8	color-alternate-bg	91
4.11.9	color-alternate-fg	91
4.11.10	color-base-bg	92
4.11.11	color-base-fg	92
4.11.12	colorize	92
4.11.13	cursor-size	92
4.11.14	hidden-layer-alpha	93
4.11.15	hud-color-frame-bg	93
4.11.16	hud-color-frame-fg	93
4.11.17	hud-color-text-bg	93
4.11.18	hud-color-text-fg	93
4.11.19	hud-style	94
4.11.20	keep-ratio	94
4.11.21	menu-color-default-bg	94
4.11.22	menu-color-default-fg	94
4.11.23	menu-color-disabled-bg	95
4.11.24	menu-color-disabled-fg	95
4.11.25	menu-color-selected-bg	95

4.11.26	menu-color-selected-fg	95
4.11.27	menu-style	95
4.11.28	music-exclude	96
4.11.29	music-file	96
4.11.30	music-filter	96
4.11.31	pixelize	96
4.11.32	system-color-bg	97
4.11.33	system-color-fg	97
4.11.34	team-color-blue	97
4.11.35	team-color-cyan	97
4.11.36	team-color-dead	97
4.11.37	team-color-green	98
4.11.38	team-color-lightblue	98
4.11.39	team-color-magenta	98
4.11.40	team-color-orange	98
4.11.41	team-color-pink	98
4.11.42	team-color-purple	99
4.11.43	team-color-red	99
4.11.44	team-color-yellow	99
4.11.45	view-color-cursor-bg	99
4.11.46	view-color-cursor-fg	99
4.11.47	view-color-map-bg	100
4.11.48	view-color-map-fg	100
4.11.49	view-style	100
4.11.50	x-wrap	100
4.11.51	y-wrap	101
4.11.52	zoom	101
4.11.53	zoom-max	101
4.11.54	zoom-min	101
4.12	Advanced settings	101
4.12.1	bench	102
4.12.2	bench-value	102
4.12.3	bin-id	102
4.12.4	bot-iq	102
4.12.5	bot-speed	102
4.12.6	check	103
4.12.7	commands-per-sec	103
4.12.8	debug-layer-id	103
4.12.9	debug-team-id	103
4.12.10	demo	104
4.12.11	dirty-read	104
4.12.12	display-background	104
4.12.13	display-console	104
4.12.14	display-cursors	104
4.12.15	display-debug-gradient	105
4.12.16	display-debug-zones	105
4.12.17	display-fighters	105
4.12.18	display-fps	105

4.12.19	display-hud	105
4.12.20	display-log	106
4.12.21	display-map	106
4.12.22	display-menu	106
4.12.23	display-mouse	106
4.12.24	display-mps	106
4.12.25	display-preview	107
4.12.26	display-progress	107
4.12.27	display-score	107
4.12.28	display-splash	107
4.12.29	display-url	107
4.12.30	executed-again	108
4.12.31	gfx-cpu-usage	108
4.12.32	gfx-debug	108
4.12.33	io-per-sec	108
4.12.34	loader-sleep	109
4.12.35	log-level	109
4.12.36	log-timeout	109
4.12.37	magic-number	109
4.12.38	memory-bazooka-eraser	110
4.12.39	memory-bazooka-size	110
4.12.40	pilot-lag	110
4.12.41	quick-start	111
4.12.42	reset	111
4.12.43	reset-config-on-upgrade	111
4.12.44	server	111
4.12.45	target-fps	111
4.13	Script hooks	112
4.14	C to Guile API	112
4.15	C functions	112

Appendix A 2005 .plan 113

A.1	Complete rewrite	113
A.2	Technologies	113
A.2.1	Script + standard C + assembly	113
A.2.2	OpenGL	114
A.2.3	CSound	114
A.3	Functionnalities	114
A.3.1	Visual enhancements	114
A.3.2	Rules enhancements	114
A.3.3	Hey, you forgot my idea!!!	116
A.4	Road map	116

Appendix B Fanfic 117

B.1	The Battle of Emberlificoted	117
-----	------------------------------	-----

Appendix C Links 121

Appendix D	GNU GENERAL PUBLIC LICENSE	123
Appendix E	GNU Free Documentation License	135
Appendix F	Indexes	143
F.1	Concept index	143
F.2	Function and keyword index	143

1 Introduction

Read this chapter to discover Liquid War 6.

1.1 In a nutshell

Liquid War 6 is a unique multiplayer wargame. Your army is a blob of liquid and you have to try and eat your opponents. Rules are very simple yet original, they have been invented by Thomas Colcombet. It is possible to play alone against the computer but the game is really designed to be played with friends, on a single computer, on a LAN, or on Internet.

An older version, [Liquid War 5](#), is available, but is not part of the GNU Project. Only Liquid War 6 is part of the [GNU Project](#), it is a complete rewrite.

The official page of Liquid War 6 is <http://www.gnu.org/software/liquidwar6/>. For more information, you can read the [Wikipedia article](#) about Liquid War.

1.2 Project status

1.2.1 What works, and what does not (yet)

As of today, the game is in beta state. It can be installed, and you can toy around with. You can even play with. It is still far from being complete as some key features are still missing.

What works:

- The whole framework is here, some functions are not implemented yet, but the bases are set up, and they are believed solid. The game is very modular, and is fully threaded. It is designed so that graphics, sound, network and bot backends can be hacked at will. It has a complete self-test suite, many debugging built-in tools, and is regularly checked with automated tools. For instance, you can check reports concerning [global references](#), [code coverage](#) and [cyclomatic complexity](#). This is not a quick hack.
- Documentation. Yes, you're reading it.
- Version 0.0.7beta is playable. Local game between humans (up to 4 players) is possible. Two bots are implemented, named random and stupid. No great players but well, they move the cursor. A new "deatchmatch" mode, different from LW5, is in place.
- Liquid War 6 already has some features which are nowhere to be found in Liquid War 5, such as multiple layers. It can be worth the upgrade.
- Maps. A number of interesting maps have already been designed (thanks to Kasper Hviid).
- The game runs natively on GNU/Linux and has been ported to Microsoft Windows and Mac OS X. Binaries are available for all those platforms. Use at your own risk. If in doubt, get the source and compile.

In the near future:

- Network play. Top-level priority. Yes, network has been promised for months (years?) and is still not there. I said "when it's done".
- Fix bugs ;) The current engine is somewhat buggy, fighters might loose the cursor, it clearly needs polishing.

In the long run:

- Write new graphical backends so that the game does not require Mesa or any OpenGL-like subsystem. The idea is to get rid of the 3D-accelerator dependency.
- Implement all the fancy 3D features, make it possible to play Liquid War 6 on a Moebius ring.
- Use the cool features of CSound to provide dynamic, contextualized sounds & musics.
- Optimize the bot algorithm, which is probably a complex AI problem.

You might be interested in checking the following URLs, which give a view on opened tasks and bugs:

- bug list: <http://savannah.gnu.org/bugs/?group=liquidwar6>
- task list: <http://savannah.gnu.org/task/?group=liquidwar6>

1.2.2 What has changed since Liquid War 5.x?

Liquid War 6 is a complete rewrite of **Liquid War 5**. The rewrite started in 2005. So a good question is “was the rewrite worth it?”...

Here’s a list of key improvements:

- appearance, global rendering quality. Call it the way you want, Liquid War 6 simply looks nicer than any previous release. Period.
- level features, including multi-layer (allowing the map designer to create bridges and tunnels), wrapping (fighters disappearing on the left can reappear on the right). Those really change the gameplay.
- deathmatch mode. Give it a try, it does change the deal ;)
- modularity, overall code quality. While this is not a user-visible change, the game is far less monolithic, therefore hacking to revamp the graphics engine, the algorithm, whatever, is easier. The situation has changed from “this is impossible to hack” to “OK, how much time can this take?”. So while one can’t promise every idea will be implemented some day, at least many more things become possible with the new codebase.

The most interesting change is still to come, and concerns network games.

Stay tuned.

1.2.3 Revision history

Liquid War 6 releases are “codenamed” after famous, historical, real or mythical characters. Here is a short revision history. For details, see the **ChangeLog** and **NEWS** files distributed with the game. Additionnally, there’s an ever-increasing “stamp” number which is incremented each time a build is done with a different source.

- 2006-12-18 : **0.0.1beta**
- 2007-09-07 : **0.0.2beta**
- 2008-01-30 : **0.0.3beta**, codename “Napoleon”, stamp 549
- 2008-09-19 : **0.0.4beta**, codename “Clovis”, stamp 756
- 2008-12-20 : **0.0.5beta**, codename “Henri IV”, stamp 1082
- 2009-01-10 : **0.0.6beta**, codename “Cesar”, stamp 1124

- 2009-10-03 : **0.0.7beta**, codename “Geronimo”, stamp 1465
- 2010-07-05 : **0.0.8beta**, codename “Attila”, stamp 1658
- 2010-08-23 : **0.0.9beta**, codename “Chuck”, stamp 2096

1.2.4 Road map

The game will probably be labelled “6.0.0” when network mode is up and running. Until then there will probably be other improvements concerning gameplay and appearance (“eye candy”). There’s a balance to keep between the major goals such as “make that network thingy work” and the very real fact that “hacking must be fun”.

1.3 How you can help

1.3.1 Help GNU

Please remember that development of Liquid War 6 is a volunteer effort, and you can also contribute to its development. For information about contributing to the GNU Project, please read [How to help GNU](#).

1.3.2 Todo list

Here’s a short list of todo items. It is probably too early to start hacking the core engine itself, for it is still under heavy development, might undergo major rewrites, and it’s hard for documentation to keep up with the reality of the code. However, there are still many things to do.

- Try the game. Play. Test. Send bug reports. Without bug reports, bugs don’t get fixed.
- Write maps. Obviously, this is something which can perfectly be delegated. Experience shows user-contributed maps are, on average, better than maps conceived by the author...
- Translate texts. Liquid War 6 uses GNU gettext, so all messages can be translated.
- ...any help is welcome.

Feel free to join the mailing-lists, this is clearly the best place to start with.

There’s also a list of opened tasks on Savannah at <http://savannah.gnu.org/task/?group=liquidwar6> which you can browse online. Maybe there’s some task for you!

Alternatively, you can contact [Christian Mauduit](#).

2 User's manual

The Liquid War 6 user's manual hopefully contains any usefull information to install the program and play the game. If you just want to enjoy Liquid War 6 without diving into map creation and programming, this is just for you.

2.1 Mailing lists

2.1.1 General discussion

The main discussion list is [<help-liquidwar6@gnu.org>](mailto:help-liquidwar6@gnu.org), and is used to discuss all aspects of Liquid War 6, including installation, development, game strategies, and whatever subject players and hackers might want to talk about, provided it is Liquid War 6 related. If you don't know on which list to subscribe, this is the one.

To subscribe to it, please send an empty mail with a Subject: header line of just "subscribe" to the `-request` list, that is [<help-liquidwar6-request@gnu.org>](mailto:help-liquidwar6-request@gnu.org).

You can also subscribe to the list using the Mailman [web interface for help-liquidwar6](#) and consult [help-liquidwar6 archives](#).

2.1.2 Announcements

Announcements about LiquidWar 6 are made on [<info-liquidwar6@gnu.org>](mailto:info-liquidwar6@gnu.org). Subscribe to it to be informed of major releases, and other significant news.

To subscribe to it, please send an empty mail with a Subject: header line of just "subscribe" to the `-request` list, that is [<info-liquidwar6-request@gnu.org>](mailto:info-liquidwar6-request@gnu.org).

You can also subscribe to the list using the Mailman [web interface for info-liquidwar6](#) and consult [info-liquidwar6 archives](#).

Please also consider reading the [latest news on Savannah](#).

2.1.3 Bugs

There is also a special list used for reporting bugs, [<bug-liquidwar6@gnu.org>](mailto:bug-liquidwar6@gnu.org). Please try and describe the bug as precisely as possible. The more accurate the description, the more chances it will get to be fixed.

While this is the standard GNU way of reporting bugs, modern SPAM standards make it very hard to filter real bug reports from junk on this list. It is more convenient to use a web interface, the URL is: <http://savannah.gnu.org/bugs/?func=additem&group=liquidwar6> and you're really encouraged to use it instead of sending emails.

Please take a look at the [bug list](#) before submitting new bugs.

2.2 Getting the game

2.2.1 Download source

Liquid War 6 can be found on <http://download.savannah.gnu.org/releases/liquidwar6/> and <http://www.ufoot.org/download/liquidwar/v6/>.

Downloading the latest file from this place, and compile it yourself on your computer with a classical `./configure && make && make install` is the recommended way to install Liquid War 6.

2.2.2 Download binaries

Some binary packages might be available. Your mileage may vary.

GNU/Linux based systems are supported, through **Debian** .deb and **Red Hat** RPM packages. There is also a Microsoft Windows installer.

However these binaries are not necessarily available for every single version of the game.

2.2.3 GIT repository

Latest work in progress versions can be obtained with **GIT**. Here's the typical command which will fetch the latest version:

```
git clone git://git.sv.gnu.org/liquidwar6.git
```

If you are behind a firewall and can't use the native GIT protocol, you can rely on the (slower) http protocol:

```
git clone http://git.sv.gnu.org/r/liquidwar6.git
```

You can browse the code online, consult log summary, and in a general manner "follow" the project on <http://git.savannah.gnu.org/gitweb/?p=liquidwar6.git> and <http://git.savannah.gnu.org/cgi/liquidwar6.git>.

2.2.4 Daily snapshots

Alternatively, you can download daily snapshots on <http://www.ufoot.org/liquidwar/v6/snapshots/>. These files should be built every day, but a `make distcheck` is done before creating them. So if the source code is, for some reason, broken, then they won't be generated. It is a good source if you want the most up-to-date code which is not completely unstable. This is typically the place to go if you want to know a reported bug is correctly fixed. Please do not rely on snapshots generated on the fly by the source content manager, daily snapshots mentioned above are better since a `make dist` has been run on them. Using a checkout is really for developers.

2.3 Installation

This section covers installation from source. Other ways of installing the program are not described here.

2.3.1 Requirements

All these libraries are mandatory to compile the game. Liquid War 6 won't compile, let alone run, without them. Some of them could probably be replaced by equivalent tools, but this would certainly require a programming effort and some changes in Liquid War 6 source code.

- **GNU Make**. Liquid War 6 might and certainly does use GNU Make extensions.
- **GNU C library**. Sounds obvious, but you need a standard C library. It happens that glibc has some rather useful extensions (yes, as of 2006, some vendors continue to offer C libraries without `snprintf...`) and Liquid War 6 might use them. In a general manner, Liquid War 6 is part of and designed for GNU. You might however manage to compile it with limited libc support, this is the case with mingw32 for instance but, do it at your own risk.

- **Perl**. Some Makefile commands require Perl. You don't need any Perl devel packages, and you can probably use any Perl 5.x version, since no fancy recent feature of Perl is used. Just plain Perl.
- **Guile**. Possibly the most required library, since Liquid War 6 is a scheme program which uses a set of functions coded in standard C. You need at least Guile 1.8.
- **GNU MP**. GMP is a free library for arbitrary precision arithmetic, required by Guile.
- **ltdl**. This library, which comes with libtool, provides a portable alternative to `dlopen` and `dlclose`. Check that you have a `/usr/include/ltdl.h` file, or install the corresponding package.
- **zlib**. Required by other libraries, but can also be used directly by Liquid War 6 to compress network messages for instance.
- **expat**. Used to read and write XML files, which contain constants and configuration data.
- **libpng**. Liquid War 6 uses libpng to read levels (maps), not to speak of other optional libraries (SDL and the rest) who need it themselves.
- **libjpeg**. Maps can also be provided as jpeg files, so libjpeg is required as well.
- **SQLite 3**. Used to handle the list of available servers.

2.3.2 Optional libraries

While all these libraries are theoretically optional (the game will successfully compile without them), you'll obviously need, for instance, one graphics backend. Otherwise, you'll simply have no display. This is not acceptable. As of today, one can reasonably consider all SDL-related libraries are required. The rest is truly optional.

- **ncurses**. Required by readline, needs to be there otherwise readline might not be detected properly on some systems.
- **GNU readline**. Used to handle input on the console. Console is not absolutely mandatory, but it's a must-have if you want to hack the game. Console unavailable does not mean you won't get anything on stdout but, the interactive script shell just won't work.
- **GTK+**. Used to display error/critical messages, so that users who launch the game by clicking on a icon (that is, not from the console) are still visually informed of important messages.
- **Mesa**. This library provides an API similar to OpenGL and enables 2-D and 3-D drawing.
- **SDL**. SDL is used to set up a working OpenGL environnement, and handle input (mouse and keyboard).
- **SDL_image**. This SDL extension is used to read textures and other graphics from disk.
- **FreeType 2**. This library is required by SDL_ttf, to draw fonts.
- **SDL_ttf**. This SDL extension is used to draw fonts. It is UTF-8 enabled.
- **libsound**. While this tool is not used yet, it is meant to be the final sound backend, as CSounds offers great power to the composer, enabling truly dynamically generated sound & music. For now Liquid War 6 tries to detect csound 4 but as the mainstream stable release is now 5 an update is needed. It will probably be updated/fixed (Liquid War 6 using csounds 5) some day, for now you can safely **not** install csound on your system and enjoy all the possibilities of the game.

- **SDL_mixer**. This SDL extension is used to allow dynamic mixing of sounds, and it also provides a builtin OGG/Vorbis file renderer.
- **libcURL**. Used to handle HTTP requests, the idea being not to re-invent the wheel but use a robust standards-compliant generic library.

2.3.3 Installing requirements using RPM/DEB packages

You might find it convenient not to install all the requirements from source, but use your favorite GNU/Linux distribution packages.

On an RPM based GNU/Linux system, a typical command (tested with **Fedora** 11) could be:

```
yum install \
make gcc glibc glibc-devel binutils \
guile guile-devel gmp gmp-devel \
libtool libtool-ltdl libtool-ltdl-devel \
zlib zlib-devel expat expat-devel \
libpng libpng-devel libjpeg libjpeg-devel \
sqlite sqlite-devel \
ncurses ncurses-devel readline readline-devel \
libGL libGL-devel libGLU libGLU-devel \
SDL SDL-devel SDL_image SDL_image-devel \
SDL_mixer SDL_mixer-devel \
freetype freetype-devel SDL_ttf SDL_ttf-devel \
libcurl libcurl-devel \
perl lcov global valgrind graphviz gv texinfo-tex
```

On a DEB package based GNU/Linux system this command (tested with **Debian** lenny) would be:

```
apt-get install \
make gcc libc6 libc6-dev binutils \
guile-1.8 guile-1.8-dev guile-1.8-libs libgmp3c2 libgmp3-dev \
libtool libltdl3 libltdl3-dev \
zlib1g zlib1g-dev libexpat1 libexpat1-dev \
libpng12-0 libpng12-dev libjpeg62 libjpeg62-dev \
libsqlite3-0 libsqlite3-dev \
libncurses5 libncurses5-dev libreadline5 libreadline5-dev \
libgl1-mesa-glx libgl1-mesa-dri libgl1-mesa-dev libglu1-mesa libglu1-mesa-dev \
libSDL1.2debian-alsa libSDL1.2-dev libSDL-image1.2 libSDL-image1.2-dev \
libSDL-mixer1.2 libSDL-mixer1.2-dev \
libfreetype6 libfreetype6-dev libSDL-ttf2.0-0 libSDL-ttf2.0-dev \
libcurl4-gnutls-dev \
perl lcov global valgrind graphviz gv texinfo texlive-full
```

Note that those requirements really depend on the exact distribution you have, package names may vary from one to another.

2.3.4 Compiling

Liquid War 6 uses **GNU Automake**, **Autoconf** and **GNU Libtool**.

Once all the requirements are installed, run:

```
./configure
make
make install
```

Liquid War 6 supports the standard `./configure --prefix=/my/path` option (in fact, it supports much more than that) so you can install the game in any directory. You do not need to be `root` to install Liquid War 6.

2.4 Extra maps

2.4.1 The extra maps package

The main package contains some maps so that you can try out the game. Still, an additional package, called `extra-maps` or `liquidwar6-extra-maps` is available, containing more maps. It really does contain many of them, including most Liquid War 3 and Liquid War 5 legacy maps, plus new Liquid War 6 maps.

2.4.2 Install extra maps on GNU/Linux and POSIX systems

On GNU/Linux systems (and possibly any POSIX unixish system) running:

```
./configure
make
make install
```

will install the extra maps on your system automatically, they will then be available in the `extra/` sub-directory when browsing maps.

The `./configure` script has a `--enable-liquidwar6` switch which will try and find automatically if there's an existing `liquidwar6` binary in the path. If there's such a binary, it will run it and ask for its `map-path` and use this value automatically.

2.4.3 Raw install of extra maps (all-platforms)

Another solution, which works on all platforms including Microsoft Windows and Mac OS X but also works on GNU/Linux, is to simply unpack the `extra-maps` package (unzip or untar) in your custom map directory, or in the system map directory. There's nothing else to do to install these maps but simply put them on your hard drive in the right directory.

Typically on an Microsoft Windows system, you would unpack the extra maps in `C:\Program Files\Liquid War 6\map\` (system directory) and on a Mac OS X system you would unpack the extra maps in `Liquid War 6.app/Contents/Resources/map/` (system directory) or `$HOME/Library/Application Support/Liquid War 6/map` (user directory). On a GNU/Linux or POSIX system you would unpack them in `$HOME/.liquidwar6/map/` (user directory).

Next time you run the game, the maps should be browsable.

If you can't see them, run `liquidwar6 --audit` and check that the place where you unpacked the files is actually searched by the binary.

2.5 Troubleshooting

2.5.1 Compilation problems

A quick survival guide:

- Check that you have all dependencies installed. Also check their version number. Double-check that you have devel packages installed, not only run-time binaries.
- Read carefully the output of `./configure`. Running `./configure > configure.log 2> configure.err` does help.
- Editing `/etc/ld.so.conf` and running `ldconfig` as root can help if some dependencies are installed in exotic places.
- Check the values of the environment variables `CFLAGS`, `LDFLAGS` and `LD_LIBRARY_PATH`.
- Try `./configure --enable-allinone`, this will disable some fancy but somewhat complicated dynamic `.so` file support, it can help if shared libraries are handled differently on your system than on a plain GNU/Linux box.

If none of these help, consider reporting a bug, or search the mailing-lists for help.

2.5.2 Check installation

Here's a check-list to ensure that your installation is correct:

- What was the output of `make install`? `make check`?
- Is the `liquidwar6` binary in your `PATH` environment variable? It might be in `/usr/games`.
- Run `liquidwar6 --pedigree`. Look at the output. Check the compilation date & time, the version number.
- Run `liquidwar6 --audit`. What do these paths look like? Are they absolute paths? Do they exist? What's there? Normally, once the game is installed, all of them should exist, and be populated with sub-directories and files.
- Run `liquidwar6 --modules`, to know which modules where compiled. You need at least one graphical module, for instance `mod-gl`, else the game won't run.
- Run `liquidwar6 --host`, this displays informations about the host system the binary has been built for.

2.5.3 Problems running the game

Now, game looks correctly installed, but you have problems running it.

- Run the game from a terminal, not from a Gnome or KDE launcher, you need to see the console output.
- In the `$HOME/.liquidwar6/` directory, you'll find two files, `log.csv` and `dump.txt`. They might contain valuable information, read them. Note that by default on Microsoft Windows this directory is `C:\Documents and Settings\<username>\Liquid War 6` and on Mac OS X it is in `/Users/<username>/Library/Application Support/Liquid War 6/`.
- Run `liquidwar6 --defaults`. This will reset all options to defaults. You might need to run this when upgrading from a version to another, since some options might appear, disappear, or defaults values can change.
- Run `liquidwar6 --test`. This should run a complete test suite, many functions in the game will be tested automatically, and errors reported.

- Run `liquidwar6 --show-script-file`. Are you really running the right code?
- Game segfaults: try `make uninstall && make clean && make && make install`. Many problems can come from using a wrong shared module, especially with beta versions.
- Game (still) segfaults: try `gdb liquidwar6`. Type `run` and watch output.
- The dynamic library loader can sometimes have problems, and does not always report explicit messages on `stdout` or `stderr`. You can change this by modifying some environment variables: `export LD_DEBUG=all`. This is very verbose but does help finding bugs.
- Consider compiling the game using `./configure --enable-valgrind` and then run it using `Valgrind`.
- Try `find / -type d -a -name "liquidwar6*" 2> /dev/null` to ensure you don't have an old version of Liquid War 6 somewhere else...

2.6 Quick start

2.6.1 Quick start

Once the game is installed, run it, click on **Quick start** with the mouse, and control the red 'a' cursor with the mouse, or keyboard, both work. Try and surround the green team, it's a stupid bot, you should win ;)

Your army is formed by all the red pixels on the screen, they should try and rejoin the cursor (the blinking 'a' letter) using the shortest path. When red and green meet, they fight. Try it, toy around.

The **Quick start** button will always make you play red against a green stupid bot, whatever other options you have set up.

Todo...

2.7 Strategy tips

2.8 User interface

2.8.1 A reduced set of keys

Liquid War 6 can be controlled using a reduced set of keys. This is to make the game more portable and allow possible ports to platforms where a full keyboard is not available. Depending on the graphics backend, exact mapping might change, they should be obvious and intuitive.

Those keys are:

- `up` : the arrow up key
- `down` : the down arrow key
- `left` : the left arrow key
- `right` : the right arrow key
- `enter` : the enter / return key
- `esc` : the escape key

- `pgup` : the page up key
- `pgdown` : the page down key

2.8.2 Combining mouse, keyboard and joysticks

It's also possible to control the game with the mouse only, or with a joystick. By default the interface will trap all events and respond on any of these possible devices.

The mouse follows these rules:

- mouse left-click is equivalent to the keyboard `enter`, it validates an item
- mouse right-click is equivalent to the keyboard `esc`, it goes back, cancels, ...
- mouse wheel-up is equivalent to the keyboard `pgup`
- mouse wheel-down is equivalent to the keyboard `pgdown`

The joystick bindings is done as follows:

- button A is equivalent to the keyboard `enter`, it validates an item
- button B is equivalent to the keyboard `esc`, it goes back, cancels, ...
- button C is equivalent to the keyboard `pgup`
- button D is equivalent to the keyboard `pgdown`

What these buttons are exactly on your joystick depends on your hardware and driver. Note that Liquid War 6 converts any analog joystick input to a digital pad-like information.

2.8.3 Quit with F10

There's also an (almost) hardcoded shortcut which will quit the game immediately, or at least as quickly as possible, without any prompt or warning.

It is the F10 key.

Think of this feature as the procrastinator's "whoops, here comes my boss!!!" emergency function.

2.9 Network games

Not implemented yet.

2.10 Graphics

2.10.1 Standard, high and low resolution

Liquid War 6 will try and pick up a default resolution when the game is launched the first time. It won't use your maximum screen resolution but will instead list all available fullscreen modes, and pick up one which is usually something like two thirds of the highest mode. This is to allow switching back and forth between fullscreen and windowed mode using the same settings. This automatically picked-up resolution really depends on your hardware and driver. It is called "standard" in the graphics options menu.

Then it is possible to automatically select the minimum and maximum resolution your hardware allows in fullscreen mode. These are called "low" and "high" in the graphics options menu. Just click on the button that display the resolution, it will change and use

the next setting. In windowed mode, the game won't accept the highest available mode but will instead use a percentage of it, defined by the `--windowed-mode-limit` parameter.

You might still be in a case where this is not enough. For instance your maximum resolution is 1600x1200, Liquid War 6 picks a default mode of 1280x960 for you but for some reason you want to play in 800x600, fullscreen. In this case, simply switch to windowed mode, resize the window with the mouse (the resolution button will show you the current resolution) and just choose a resolution near 800x600. It does not even need to be exactly 800x600, 798x603 would probably fit. Then when switching back to fullscreen, you'll be in 800x600, the game will automatically pick up the fullscreen mode which is closest to the current windowed mode resolution.

2.10.2 Display rate

By default the game will try and run at 60 frames per second. Given the nature of Liquid War 6, this is probably enough. Higher values will maybe give a slightly smoother display, but barely noticeable.

You can activate the display of frames per seconds (aka "fps") through the menu ("options -> system") or with the command line ("`--display-fps`").

On a single processor system, reducing the number of frames per second might allow the rest of the game run faster. So if you notice the game is really slow, in terms of "fighters move slowly" then you might be happy reducing the display rate and therefore giving power back to the other parts of the program. On a dual-core (or more) or on a multi-processor system, this is probably useless since the game is threaded and has a dedicated thread for display purposes. The command line option to reduce the number of frames per second is `--target-fps`.

Additionally, the parameter `--gfx-cpu-usage` allows you to force the display thread to "take a rest" and go idle for some time. This is advanced settings, most users won't touch this.

2.11 Sound & music

2.11.1 Current status

As of today, the game is capable of playing **Ogg Vorbis** audio files. That's it.

2.11.2 The future

In the long run, what is planned is to support **Csound** which would allow very cool effects, such as dynamically changing the music while the game is running, typically following the action. If there's a lot of fight, the music could reflect this.

For now this is only vaporware, just a nice idea among others, nothing implemented yet.

2.12 Config file

The config file is a simple XML file. It uses XML only to benefit standard parsing tools, but it's not a structured XML file, in the sense that the tree is so simple that all items are at the same level. It is just a simple key-value binding.

This file is in `$HOME/.liquidwar6/config.xml` on GNU/Linux and POSIX systems, in `C:\Documents and Settings\<username>\Liquid War 6\config.xml` on Microsoft

Windows and in `/Users/<username>/Library/Application Support/Liquid War 6/config.xml` on Mac OS X.

You're free to edit it manually, but all parameters are changeable with command line options. The program will overwrite this file each time it exits, so if you put comments in it, they will disappear. The advantage of this is that if you misspell something, or if for some reason the game does not understand a value, then when rewriting the file, it will show you it just did not get it.

The file embeds the documentation for all its entries, it is therefore rather verbose. The documentation is the same you will find online or by quering the game with the `--about` option, also the same you would get reading this manual.

2.13 Logs

Liquid War 6 uses `stdout` to output important messages, and `stderr` to log warnings and errors. It will also use `syslog` if available.

Additionally, a verbose log is available in `$HOME/.liquidwar6/log.csv` on GNU/Linux and POSIX systems, in `C:\Documents and Settings\<username>\Liquid War 6\log.csv` on Microsoft Windows and in `/Users/<username>/Library/Application Support/Liquid War 6/log.csv` on Mac OS X.

You can read this using any spreadsheet software capable of reading csv file. It uses the tab (`\t`) character as a separator. It contains valuable informations including version and most default values for the game, and for each line logged, it says where in the code the log function was called. A must-have for debugging.

2.14 Report bugs

There are two ways to report bugs:

- send a mail to [<bug-liquidwar6@gnu.org>](mailto:bug-liquidwar6@gnu.org);
- use the web-based Savannah bug tracker: <http://savannah.gnu.org/bugs/?func=additem&group=liquidwar6>.

The latter (**Savannah**) is much preferred, because the mailing-list is bloated with spam... It also offers a **list of bugs** which you should read before submitting a new one.

3 Hacker's guide

This hacker's guide is for anyone who is curious about the game, and wants to know how it works. It covers many aspects from simple map creation to technical program internals. A great effort has been done in Liquid War 6 so that it should be much more hackable than previous versions. Any feedback is welcome.

3.1 Designing levels

3.1.1 Why is level design so important?

As of **Liquid War 5**, most levels have been contributed by players. While the maintainer of Liquid War 6 has technical knowledge to develop the game, artistic talent and taste might not be his domain of excellence 8-)

Therefore contribution are truly welcomed when they take the form of a new, original, fun and good looking level. It's believed the levels often make the game much more than its engine. This is true for any type of game, and Liquid War is no exception.

So this section is here to help players understand how to hack existing levels, and create new ones, in the hope that 1) they can enjoy their own creations and 2) possibly share their work with others.

Note that this manual might refer to levels and maps: they are just two different names to describe the very same thing. It's an alias.

3.1.2 Format overview

Liquid War 6 stores level information in a plain directory.

There is no such thing as an opaque `.dat` binary file. The name of the level is the name of the directory itself, and its elements are the files contained in it.

Files must follow a precise naming scheme. For instance Liquid War 6 expects a `map.png` file to be present in each map directory.

All image files in a level use the **Portable Network Graphics** or **JPEG** format. It is possible that in the long term, Liquid War 6 will be able to handle levels as `.tar.gz` or `.zip` files. In that case these files will only be a compressed image of the actual level directory.

See the `./map/` directory of the source Liquid War 6 distribution to see example of maps.

3.1.3 `map.png`

This is the only required file in a level.

In fact, the existence of `map.png` makes a directory a level. When checking whether a directory is a correct level, Liquid War 6 simply tests the existence and validity of `map.png`.

This image is a simple black & white area, where white zones are the background, the sea, the places where fighters can move, and black zones are the foreground, the walls, the places where fighters can't go.

This informations can be stored in a 2-color indexed file, or in a grayscaled or even truecolor RGB file, but color information won't be used. Internally, Liquid War 6 will read the color of every point. If it is over 127 on a 0 to 255 scale, it will be considered as background, if it is below 127, it will be considered as foreground.

3.1.4 layer2.png ... layer7.png

Liquid War 6 can handle mutiple layer maps. Think of a pile of maps, one being on top of the other. This allows you to create a volume, the game considers every layer has two axis x and y, and the z axis is to travel through layers. First layer corresponds to z=0, second layer to z=1, and so on.

Here are the files you can use to define layers:

- `map.png` this one is on top, it's always defined (z=0)
- `layer2.png` (z=1)
- `layer3.png` (z=2)
- `layer4.png` (z=3)
- `layer5.png` (z=4)
- `layer6.png` (z=5)
- `layer7.png` (z=6)

A `layerX.png` file should be designed exactly like `map.png`. In fact, `map.png` could simply have been called `layer1.png`.

Up to 6 extra layers can be defined (from `layer2.png` to `layer7.png`). This is a hard-coded limit. It allows you to define 7 different layers, including the top `map.png` layer. Keep in mind this layer system is not real 3D, it's more a "2D and a half" model. Adding layers can considerably slow down the game, so it's wise to try and use as few layers as possible. Technically, 3 layers will allow you to build bridges and tunnels, which is probably the most usefull construction using layers. Fighters can also have difficulties navigating through layers so piling up layers in narrow "vertical" z-axis based tunnels is probably not a great idea.

The `ufoot/concept/pass` map of the `liquidwar6-extra-maps` demonstrates basic layer usage.

3.1.5 texture.png, texture.jpeg and texture-alpha.jpeg

It is possible to define a texture for the map by putting a `texture.png` or `texture.jpeg` file. It does not need to have the same dimensions as the map itself. Indeed, textures can be much more precise than the actual logical map.

There's no theoretical limit on how big a texture can be, more precisely, it can be much bigger than any hardware/driver maximum texture size. In practice, a too big texture will waste your video card RAM, and slow everything down. Sizes ranging from 640x480 to 1600x1200 are reasonable texture sizes.

If you don't define this, the `map.png` file will be used as the texture, and also import colors from `style.xml` if defined.

Note that the shape of the texture defines the shape of the map, that is, the ratio with which it will appear on the screen.

The PNG alpha layer will be used for transparency. But to save disk space, it can be convenient to prefer the JPEG format, use `texture.jpeg` instead of `texture.png` and store the alpha layer in a separated file, called `texture-alpha.jpeg`. This avoids handling heavy PNG files, PNG compression not being performant on most textures.

In `texture-alpha.jpeg`, black is considered opaque, and white is transparent. Different levels of gray correspond to different levels of opacity. Previous versions of the game used the other way of doing things (black is transparent) because this is technically, the most obvious way to do things. Black is 0 and transparent is 0. But for a human “reader” of the map this does not make sense. One generally expects white to be the equivalent of “undrawn” or “blank”, well, if it's undecided, void, transparent, whatever, it's white. When **the Gimp** flattens an image, it becomes white, not black.

So white is transparent. Period.

3.1.6 `glue.png` and `boost.png`

If there's a `glue.png` or `boost.png` file in the map directory (you can use one of them or both) then they will be interpreted as follow:

- on areas where `glue.png` and `boost.png` are white, nothing special happens, fighters follow their default behavior
- on areas where `glue.png` is black, fighters will be slowed down. How slowish they will be depends on the 'glue-power' parameter. If 'glue-power' is 3 then fighters will move three times slower.
- on areas where `boost.png` is black, fighters will behave faster. How fast they will be depends on the 'boost-power' parameter. If 'boost-power' is 2 then fighters will move two times faster.
- on areas where `glue.png` or `boost.png` are gray, they will be slowed down less or speeded up less depending on how dark the grey is.

There can be, at the same place, some gray or black in both `boost.png` and `glue.png`. How this will behave exactly is not really clear at this stage, the recommendation is not to do this (it does not really make sense anyway) but if you do it, game won't complain.

It's also wise not to abuse of `boost.png` for obviously, a map filled with “boosted” zones at a X10 pace will require much more CPU than the same map with no such setting. This might fool the automatic resampling algorithm and lead to maps that are unplayable. The spirit of `boost.png` is just to make a few spots go faster.

It's also important to note that behaving faster or slower means moving faster or slower but also attacking faster or slower, and, in a general manner doing any action with a different pace.

3.1.7 `danger.png` and `medicine.png`

If there's a `danger.png` or `medicine.png` file in the map directory (you can use one of them or both) then they will be interpreted as follow:

- on areas where `danger.png` and `medicine.png` are white, nothing special happens, fighters follow their default behavior
- on areas where `danger.png` is black, fighters die automatically, that is, they become black and loose health. How dangerous these zones are depends on the 'danger-power' parameter.
- on areas where `medicine.png` is black, fighters regenerate faster, they become bright and shiny as if auto-healing. How efficient this medicine is depends on the 'medicine-power' parameter.

- on areas where `danger.png` or `medicine.png` are gray, well, it's in between, the “danger” and “medicine” effect will be proportional to the level of gray.

There can be, at the same place, some gray or black in both `medicine.png` and `danger.png`. How this will behave exactly is not really clear at this stage, the recommendation is not to do this (it does not really make sense anyway) but if you do it, game won't complain.

3.1.8 one-way-<direction>.png

The four files:

- `one-way-north.png` (AKA “up”)
- `one-way-east.png` (AKA “right”)
- `one-way-south.png` (AKA “down”)
- `one-way-west.png` (AKA “left”)

can be used to force the fighters to go in one given direction, on some parts of the map. If an area is black on one of this meta-layers, then fighters will go in the given direction. For instance, a black zone in `one-way-north` will make fighters go to the north (AKA “up” direction) regardless of the cursor position. The fact that this is a one-way path is understood by fighters and they will take this in account when choosing the shortest path to go somewhere. You can combine vertical and horizontal one-way informations, making diagonal one-way paths.

3.1.9 rules.xml

Whereas `style.xml` is only about the appearance of the map, `rules.xml` allows the map designer to change pretty much any parameter.

Ultimately, the player can still ignore these settings and override them with its own values, but the idea is: most game options are only pertinent in a given context. For instance, on some maps it's interesting to move slowly, on some other it's interesting to move fast. Some maps might be playable packed with fighters everywhere, some other might be much more fun with almost nobody on them.

The approach in [Liquid War 5](#) was to make the options available, but let the player himself find the right settings for the right map. The consequence is that no one ever used all those cryptic options in the advanced options menu, and probably 99% of the players ended up playing with default settings. This is not that bad, but given the fact that changing a few parameters one can totally transform the gameplay, it has decided been that in Liquid War 6, the map designer suggests the right options that matches his map.

This does not prevent the player from toying with options himself, he can still do it.

There's also one important point to note: all these options are technically implemented as integer parameters. We certainly do not want any float here, since, and it is a Liquid War specific behavior, the game must be 100,00% predictable and behave the same on every platform. As there is nothing like exactness when speaking of floats, those are forbidden here. As for strings, we are dealing here with low-level internals, and this section is not about telling a story. They are technical options only. Booleans are implemented with the usual `false = 0` and `true = 1` convention. Note that other config files in Liquid War 6

might rely on floats, strings, and booleans with conventionnal `true` and `false` values, but not this one. `rules.xml` is special.

This `rules.xml` file is a direct image of the internal “rules” structure, so it contains technical, sometimes not very user-friendly parameters. While hacking `rules.xml` directly is a good way to test things, most of the time, the other file `hints.xml` contains more high-level informations that do the job the right way. A typical example is speed.

See [Section 4.9 \[Map rules.xml\]](#), page 68.

3.1.10 hints.xml

This parameter is only used by the map loader. The map itself contains none of these parameters, they are only clues (hints, in fact..) on “how to load the map” which are passed to the loader.

Let's take an example : speed. This `rules.xml` file has a (rather) easy to use “speed” parameter, which will do all the job of finding the right resolution for your map, the right “rounds-per-sec” and “moves-per-round” parameters, in short, it will set many other parameters to fit your needs.

As far as the map designer is concerned, `rules.xml` and `hints.xml` could have been merged (but so would have `style.xml`) but internally they are very different: `rules.xml` contains the real parameters, the one used by the algorithm whereas `hints.xml` contains only instructions which are used once when loading the map and then disappear. The core algorithm has no idea of what was in `hints.xml`, once it's loaded.

See [Section 4.10 \[Map hints.xml\]](#), page 83.

3.1.11 style.xml

This is a simple XML file defining various appearance parameters. It has absolutely no effect on gameplay. These settings can ultimately be overridden by the player, but the idea is that if the map designer thinks this level looks better with this or that option, let him say it in this file.

See [Section 4.11 \[Map style.xml\]](#), page 89.

3.1.12 Resampling

This is a very important point. Liquid War almost **always** resamples maps, unless you ask it not to do it. This is not recommended, it is believed in the general case, letting the internal algorithm make its own decisions is better than trying to figure out oneself “which is the best resolution”.

The reason is, the right resolution (we're talking here of the logical resolution, how many fighters wide is the battlefield) often depends on the speed and general resources of the computer the program is running on. The map designer does not have this information. The program does. It runs a bench at startup. So this way it can choose, at runtime, the resolution which fits best.

The recommended way of doing things is not to try to be too picky about `rules.xml` parameters related to speed and also let the default map size limits in `hints.xml` to their defaults. Do not use them unless debugging stuff. Then the program will resample the map so that the player can play on it at a reasonable speed. If map is too big, and it's often the case, then it will downsize it until there are sufficiently few fighters so that the CPU can

handle the job. This, of course, is not rocket science. The bench calculation is a somewhat brute-force approach of doing things. Formally, we would have to run the map for good to figure out what is the right speed. Still, this bench gives good approximations.

Previous versions of the game relied heavily on 'fighter-scale' to resample maps, but this is not the case anymore. The 'fighter-scale' is now a minor parameter which is used to upsize maps if they are too small. In 99.9% of the cases, the map is first upsized by 'fighter-scale' for this parameter is by default set low (1.0) then downsized by 'bench-value' for real-life personal computers can't handle 1600x1200 maps in real-time. Not yet.

There are a bazillion options to control map size, including 'min-map-surface'. They are here because it's important that, ultimately, people can do whatever they want with the game. But for map design, this is another story. Don't use them. Rely on 'bench-value' and just care about game speed. This is achieved by changing the "speed" parameter.

3.1.13 Music

It is possible to store your own custom music file within the map directory. You can call it whatever you want (you can keep its original name, which is something music authors usually appreciate, even if there's no strong "attribution" clause on the license, it can be considered fair use not to fiddle too much with the name) you just have to place it in the same directory than the other files like `map.png` or `texture.jpeg`.

The following formats are known to work with the default `SDL_mixer` based `mod_ogg` backend:

- `ogg` (`Ogg Vorbis` files)
- `wav`
- `midi` (extensions `.mid` and `.midi` should both work)
- `mod`, `s3m` and `xm` files, AKA "modules".

To be more precise, here's how things work:

- step 1: the game tries to find the file `music-file` (parameter taken from `style.xml` or defined/overridden by player) in the current map directory; step 2: if not found, it will try every path in `music-path` to find this file. This includes the "system" music directory with musics that ship with the game, but also the `./music` subfolder in the user directory; step 3: if still not found, it will try to play a random file, relying on `music-filter` to ignore some files.

3.2 Translating

3.2.1 Using gettext

Liquid War 6 uses `GNU gettext` for all its messages. There's an [online manual](#) about this tool. In practice, what you have to do as a translator is to edit the `po/xx.po` file with `xx` being your language / country code. For instance, to translate the game in French, one needs to edit `po/fr.po`.

3.2.2 Formatted strings

This is very important, you might already be aware of it if you are familiar with `gettext`, but still it's worth mentioning : when a string contains special characters such as `%d` or `%s`

(in a general manner, anything with a % it's important that all translations contain exactly the same number of %ds and %ss than the original.

For instance:

```
"foo has %d bars (%s)"
```

can be translated to:

```
"ziblug zdonc %d zuc - %s - tac"
```

The number, order and type of % entries is preserved. To learn more about these formats, use `info printf` or `man 3 printf`. In a general manner, get informations about [printf](#).

Additionally, some strings are used by Scheme (Guile) code and not by C code. Thus, they don't use the standard C/printf convention. In these strings, what you must preserve and be aware of is the tilde character ~. Very often you'll see ~a in a string. As with the printf %, you must preserve the number, order and type of those. There is a complete [online reference](#) about this way of formatting strings.

3.2.3 Partial translation

Liquid War 6 has thousands and thousands of messages which could theoretically be translated. In practise it's counter-productive to spend time to translate those, as the game is still evolving constantly, and as most of these messages are technical messages which inform about rare bugs and strange conditions. All sort of informations which, while valuable, are not intended for end-users and are more destined to be reported in bug reports. To select only the interesting messages to translate, the current gettext configuration only uses a reduced set of files.

- `src/scriptpo.c` : the most important file. It contains the definitions used by all the Guile code, this is where you'll find all the menu labels.
- `src/lib/sys/sys-log.c` : log messages and keywords. These are not the log messages themselves, it only concerns the log engine. One can for instance replace `WARNING` by `ATTENTION`.
- `src/lib/hlp/hlp-credits.c` : the credits, which are displayed at game startup in the splash screen.
- `src/lib/lw6-print.c` : contains some messages printed on the console.

As a side note, the file `src/lib/hlp/hlp-reference.c` contains all the entries for the various configuration options, anything that can be queried by `liquidwar6 --about=<keyword>`. This is several hundred messages. It might be interesting to translate them some day, but it's obviously not a priority today.

3.3 Architecture

3.3.1 C + Guile

Technically, Liquid War 6 is a collection of C functions which are exported to Guile. The main binary embeds a Guile interpreter, which will run a Guile script. This script calls the exported C functions, and glues them together.

It should be possible to implement the game without using Guile at all, using C code to make the various modules communicate together. This might seem an easier way to go,

not involving several languages. However, using this script level is a good way to achieve several important goals:

- it's possible, at any time, to query the game about its internal state, dump objects, take actions. That's what the console is about. It's a bit like having an embedded debugger, it's really a very convenient tool to develop, make experiments and track problems.
- many hacks can be done without recompiling anything at all. Simply edit a few files with an editor, and your patch is running. Once the binary base is set up, hacking scripts on top of it is (almost) a piece of cake.
- forcing the program to use scripts to transfer informations from a module to another is a good way to avoid "spaghetti" code, when modules cross-use each other in an uncontrollable way. Of course in some cases, modules communicate directly, especially when performance is important. But for many tasks, it's just very comfortable and safe to have module A send orders to module B through a high-level script API.

Having Guile to implement high-level stuff also decreases, to some extent, the need for object-oriented features of C++. The big picture is : low level code that require speed, optimized data processing, is written in C. Code which is more high level and requires abstraction is written in scheme.

3.3.2 Threading and SMP

Liquid War 6 makes a heavy usage of threads. Early versions of the game did not have this feature but starting with 0.0.7beta, one can really consider the game is heavily threaded.

There's basically:

- a thread to handle the main control flow. This thread runs scheme code which Guile. It's not the most CPU-greedy thread, but when it's stalled, there's no more interaction between the user and the program.
- a thread to handle the display. Depending on rendering options, this thread can consume lots of CPU cycle. On a single processor/core system, it can be interesting to lower rendering options in order to gain speed on other aspects of the game. On a quad-core system, it's probably useless, just play with all bells and whistles activated.
- two threads to run the core algorithm. One maintains the so-called reference state, the other being dedicated to the draft state. In a local game there's no draft state so only one of those two threads is used. There's even a technical optimization which can be turned on and can theoretically use even more threads and be efficient on very big maps but well, it's rather untested and still has to prove its real efficiency.
- a thread to handle map loading. This one is not active all the time, it's just here to keep a preemptive interface while loading complex maps.
- network code can also fire threads, especially when connecting on remote systems.

So globally, if you have an SMP system, the game will be happy with it. It will also run on a single processor, as the program uses POSIX pthreads it's capable to run on any computer which has pthreads implemented for it.

But, and this is a strong limitation, without pthreads, the game won't run. At all. Or at least, not unless it's almost completely rewritten.

3.3.3 Internal libraries

The C code is splitted into several internal libraries. This allow independant testing of various game modules.

The main module, the most important one, is `ker` (stands for “kernel”). This is were the core algorithm is. To some extent, the rest of the code is just about how to provide this module with the right data and environment. Logically, if you profile the game, you should find out that a great part of the CPU time is spent here. Code here is about spreading gradients, moving fighters and cursors.

The `map` module is here to handle maps, it contains the code to manipulate maps in memory. But it does not know how to load them from disk. This is the responsability of another module, `ldr`, which is linked against libraries such as `libpng` or `libjpeg` and does the job of transforming those standard formats into a usable in-memory structure. There's still a third moduled involved in map handling, it's `tsk`, whose job is to load a level in the background. It has a 2-steps asynchronous loading system which allows the game to load maps while the user interface is still responsive, and give a preview of the map as soon as possible, when loading continues in the background, building optimizing structures which are usefull when playing but not mandatory just to show the map.

At the other end of the algorithm-chain, the `pil` module will “pilot” things. It's this module which will translate text readable orders (typically adapted for network usage) into function calls. It has event lists, keeps them in the right order, and will also permanently maintain three different states of the game. A backup state which can be used any time to go back in time and get the game in a stable 100% sure state. A reference state which is correct but ever changing. Basically backup plus all the orders received between backup and reference gives reference. And finally a draft state which is as up to date as possible but might be wrong. This is typically interesting in network game, where we want to show something moving, something fast, even if there's lag on the network and other computers fail to send information in time. In this case we display draft while still keeping reference and updating it when we finally receive valid informations. Backup would be used to send bootstrap information when people are joining a new game, or to check up if things are going right.

A special `bot` module is here to handle bot algorithms. A bot is just a simple `move` function which takes a game state as an input, and returns an `x,y` position, just the way a mouse handler would. How complex a bot is “under the hood” depends on the type of bot. Current bots are really basic.

The `gfx` module handles all the graphics stuff. It is itself splitted in several sub-modules, that is, it does not do anything but load a module such as `mod-gl` which will actually contain the implementation. In an object-oriented language, it would be an abstract class, an interface. The implementation does not need to be thread-safe. It's better if it is, for theoretically it could be possible to fire Liquid War 6 with two display backends running at the same time on the same game instance, but this code has yet to be written, and it's a rare dual headed configuration which probably has no real-life usage. If only one graphics backend is activated at a time, the rest of the implementation guarantees there will never be two concurrent calls to a function in this module. It is the `dsp` (“display”) which handles this. It fires a thread for rendering purposes, and sends information to this thread, detecting automatically if it's necessary to acquire a mutex and update rendering informations. For

the caller, this is transparent, one just has to call an update function from time to time. The module will even perform “dirty-reads” on a game state being calculated, to render things in real time, as soon as possible.

To ease up the implementation of different graphics backends, a `gui` module contains code which is meant to be used by any graphics backend. It’s just a factorisation module, containing common code and interfaces, related to displaying things. This is where, for instance, one can find a high level menu object.

The `snd` module handles sound. It’s also an abstract class, an interface, which uses dynamic backends as implementations.

The `net` module is a wrapper over different network APIs, it handles Winsock and POSIX sockets in a uniform manner. The `cli` and `srv` contain network client and server code, implementing the various protocols in dynamically loadable sub-modules. It’s the role of `p2p` to glue this together, handle the list of available servers, the message queue, verifying nobody is cheating, and so on. All this modules share information about current game state using code & structures defined in `nod`, and use message utilities (format, parse) defined in `msg`.

The `sys` module contains most system and convenience functions, it handles logs, type conversions, timer, memory allocation, it’s the fundamental module every other module depends on. It has a compation `glb` module with all the `Gnulib` shared code.

The `hlp` is used to handle keywords and internal self-documentation (this is what is used by `--list` and `--about`), `cfg` knows how to read and save config files, `cns` handles the console, and `dyn` can load `.so` shared files dynamically.

Finally there are small modules like `img` (to make screenshots of the game) which have been separated because they required special libraries to link with and/or did not really fit in existing modules for dependencies reasons.

So well, this is a lot of modules. The list might move a bit, but the big picture is here. Each module is testable separately.

3.4 Memory structures

The most important memory structures in Liquid War 6 are:

- `map (lw6map_level_t)` : this contain the map immutable informations. This is what resides in memory after a map has been loaded from the disk. It contains all the various `.png` and `.jpeg` files stored as pixel arrays, resampled if need, and also contains the various map attributes. Once this structure is ready, the game is capable of displaying the map on the screen, but it can not do anything with it yet.
- `game_struct (lw6ker_game_struct_t)` : this one contains the same informations as the previous structure, only the information has been post-treated so that it’s ready for use by the core algorithm. It will, for instance, contain the famous mesh structure, which groups squares by packets of 1, 4, 16, 64 or more. The reason it’s been separated from the level is that operations such as creating the mesh might require a lot of time. So to allow players to see the level while black magic is still running in the background, it was required to make a difference between what is required to view the map (“level”) and what is required to play on it (“game_struct”).

- `game_state` (`lw6ker_game_state_t`) : contains all the variable, ever changing game data. This is where the position of fighters is stored, their health, and such things. It is designed to be synchronizable by using mostly simple calls to `memcpy`. It heavily relies on the previous structures, the idea is that one can have several “game_state” plugged on a single “game_struct”.

All these structures are defined in the `ker/ker.h` header.

3.5 100% predictable algorithm

The core Liquid War 6 algorithm is 100% predictable, that is to say, given the same input, it will produce the same results, on any computer. Previous versions of the game also had this property. This is very important for network games, since in a network only informations such as “cursor A is at position x,y” are transmitted. Every node maintains its own internal game state, so it's very important that every node comes with the same output given the same input.

For this reason Liquid War 6 never uses floating point numbers for its core algorithm, it uses fixed point numbers instead. It also never relies on a real “random” function but fakes random behavior by using predictable pseudo-random sources, implementation independent, such as checksums, or modulus.

There are also some optimizations which are not possible because of the predictability requirement, for instance one can not spread a gradient and move the fighters in concurrent threads, or move fighters from different teams in different threads.

If you read the code, you'll find lots of checksums here and there, a global checksum not being enough for you never know where the problem happened. The test suite uses those facilities to guarantee that the game will run the same on any platform.

Not being able to rely on a predictable algorithm would require to send whole game states on the network, and this is certainly way too much data to transmit. A moderate 200x200 map has a memory footprint of possibly several megabytes, so serializing this and sending it to multiple computers at a fast-paced rate is very hard, if possible at all, even with a high bandwidth. We're talking about Internet play here.

3.6 About mod-gl

3.6.1 The main renderer

Liquid War 6 has a modular architecture which allows the programmer (and the player) to plug pretty much any rendering/graphics backend, provided this one is... developped.

As of 2009 the only available backend is still `mod-gl`, it will display the game using 3D acceleration, if available, through the **SDL** library, using its GL bindings.

Additionally, versions available for Microsoft Windows and Mac OS X will probably never any other backends available. For technical reasons, those platforms do not have the flexibility of GNU/Linux and do not allow graphical libraries to be loaded dynamically. In practice, both of them require hacks that override the standard `main` function. Microsoft Windows has its `WinMain` instead, and Mac OS X is even more pedantic, requiring graphical functions to be executed in the main thread. So `mod-gl` is just linked statically in those versions, and the modularity of the game is purely theoretical on these platforms.

This `mod-gl` module is really one of the key stones of Liquid War 6, and if you want to change graphical things, it's definitely the place to hack on. The source is in `src/lib/gfx/mod-gl`.

3.6.2 Hardware requirements

The `mod-gl` backend requires “moderate” hardware, but it still does require hardware acceleration. Pure software rendering through `mesa` for instance, won't be enough.

So if you're running Xorg on GNU/Linux and there's a DRI driver for your card, the game should run fine.

On the programmer side, the counterpart is that one should not rely on fancy OpenGL features. Textures have a maximum size of 512x512 for instance. Of course some maps are bigger than this but this means that internally, `mod-gl` splits them into smaller tiles, and displays those tiles one by one.

3.6.3 The gl-utils toolbox

Inside the `mod-gl` backend, the `src/lib/gfx/mod-gl/gl-utils` directory contains lots of common structures, factorized functions which can (and should, if applicable) be used.

3.7 Compilation tips

3.7.1 Advanced `./configure` options

In addition to all the common `Autoconf` switches such as `--prefix`, Liquid War 6 has some custom switches:

- `--enable-console`: allows you to turn on/off console support. Normally this is detected automatically but in case you really want to disable it on platforms which support it, you can. This will cause the program not to link against `libreadline`, among other things.
- `--enable-gtk`: allows you to turn on/off gtk support. Normally this is detected automatically but in case you really want to disable it on platforms which support it, you can. This will cause the program not to link against GTK libs.
- `--enable-optimize`: will turn on optimizations. This will turn on compiler options such as `-fomit-frame-pointer` but also disable some code in the program. Indeed, most of the advanced memory checking in the game - which ensures it does not leak - will be turned off. This will certainly speed up things, however, it's not recommended to turn this on until program is not stable enough so that memory leaks and other problems can be declared 'impossible'. Turn this on if you really have some speed problem, otherwise it's safer to use the full-featured 'slow' version of the game.
- `--enable-paranoid`: will turn on very picky and pedantic checks in the code, try this when you suspect a serious memory bug, a race condition whatsoever, and want to track it down. Useless for players.
- `--enable-headless`: will allow compilation without any graphics backend. The game is unplayable in that state but one can still wish to compile what is compilable, for testing purposes.
- `--enable-allinone`: will stuff all the internal libraries into one big executable. Very convenient for profiling. The major drawback is that you need to have all the optional

libraries installed to compile all the optional modules. Another side effect is that with this option there's no more dynamic loading of binary modules, so if your platform has a strange or buggy support for `.so` files, this option can help.

- `--enable-fullstatic`: will build a totally static binary, that is using the `--static` option for `gcc` and the `-all-static` option for `libtool`. Currently broken, this option could in the future allow for building binaries that run pretty much everywhere, without requiring any dependency but a Kernel.
- `--enable-gprof`: will enable profiling informations. This will activate `--enable-allinone`, else you would only track the time spent in functions in the main `liquidwar6` executable, and exclude lots of interesting code contained in dynamic libraries.
- `--enable-instrument`: will instrument functions for profiling. This will turn on the `-finstrument-functions` switch when compiling, so that the hooks `__cyg_profile_func_enter` and `__cyg_profile_func_exit` are called automatically. Then you can link against tools like `cprof` or `FunctionCheck`.
- `--enable-profiler`: will enable `Google Performance Tools` support. Basically, this means linking against `libtcmalloc` and `libprofiler`. You could activate those by using `LD_PRELOAD` or by using your own `LDFLAGS` but using this option will also make the game tell you if `CPUPROFILE` or `HEAPPROFILE` are set when it starts. The `pprof -gv` output is very handy.
- `--enable-gcov`: will enable coverage informations, to use with `gcov` and `lcov`. This is for developers only. It will activate `--enable-allinone`, else there would be some link errors when opening dynamic libraries. The obtained information is available online: `coverage`. and `GNU global`.
- `--enable-valgrind`: will enable some `CFLAGS` options which are suitable for the use of `Valgrind`, to track down memory leaks and other common programming errors. Use for debugging only, usually together with `--enable-allinone`.

3.7.2 Microsoft Windows msys/mingw32 port

This section describes how to compile the game from source under Microsoft Windows. Note that players are encouraged to use a free system such as GNU/Linux, which is the platform Liquid War 6 is being hacked on by default. If you encounter problems with this port, you'll probably save time by installing a double-boot with GNU/Linux coexisting with your previous Microsoft Windows install.

Basically, Liquid War 6 requires `MinGW`. More precisely, it requires `MSYS`. A standard `Cygwin` installation won't work, because it is too UNIXish to allow third party libraries like `SDL` to compile natively. You might argue that `SDL` is available for `Cygwin`, but in reality, the `Cygwin` port of `SDL` is a `MinGW` port. Indeed, `Cygwin` brings all standard POSIX functions including the use of `main` instead of `WinMain` and I suspect this is a problem for graphical libraries like `SDL` which do require some sort of direct access to the OS low-level functions. Therefore, `MinGW` is more adapted for it does not define all these functions, and allows any library to hook on Microsoft Windows internals directly. Point is then, you also lose the cool effect of `Cygwin` which is to have a complete `glibc` available, including network functions like `select` defined the POSIX way, and not the `WinSock` way. If you ever ported code from POSIX sockets to `WinSock 2`, you know what I mean. Using `MinGW` is also embarrassing for some libraries won't compile easily, and for instance programs which

heavily rely on a real TTY interface to work are usually hard to port. This includes [ncurses](#) and [GNU readline](#). Liquid War 6 tries to have workarounds for all this, and in some cases the workaround is simply that embarrassing code is not compiled on Microsoft Windows. For this reason, some features are not available on this platform. Period.

Now the reason you need MSYS and not only MinGW is that MSYS will allow `./configure` scripts to run, and this eases up the porting process a lot. MinGW and MSYS packages are downloadable on the [SourceForge MinGW download page](#). Alternatively, there is a [mirror on ufoot.org](#), but files might be outdated.

To compile Liquid War 6, first download and unzip all the following files in the same directory, for instance `C:\MSYS`. If you do not have any tool to handle `.tar.gz` and `.tar.bz2` files under Microsoft Windows, which is likely to be the case when MSYS is not installed yet, you can untar these on any GNU/Linux box, then upload the whole directory to the target Windows host.

- `autoconf2.5-2.61-1-bin.tar.bz2`
- `autoconf-4-1-bin.tar.bz2`
- `autogen-5.9.2-MSYS-1.0.11-1-bin.tar.gz`
- `autogen-5.9.2-MSYS-1.0.11-1-dev.tar.gz`
- `autogen-5.9.2-MSYS-1.0.11-1-dll25.tar.gz`
- `automake1.10-1.10-1-bin.tar.bz2`
- `automake-3-1-bin.tar.bz2`
- `bash-3.1-MSYS-1.0.11-1.tar.bz2`
- `binutils-2.18.50-20080109-2.tar.gz`
- `bison-2.3-MSYS-1.0.11-1.tar.bz2`
- `coreutils-5.97-MSYS-1.0.11-snapshot.tar.bz2`
- `crypt-1.1-1-MSYS-1.0.11-1.tar.bz2`
- `csmake-3.81-MSYS-1.0.11-2.tar.bz2`
- `cvs-1.11.22-MSYS-1.0.11-1-bin.tar.gz`
- `diffutils-2.8.7-MSYS-1.0.11-1.tar.bz2`
- `findutils-4.3-MSYS-1.0.11-1.tar.bz2`
- `flex-2.5.33-MSYS-1.0.11-1.tar.bz2`
- `gawk-3.1.5-MSYS-1.0.11-1.tar.bz2`
- `gcc-core-3.4.5-20060117-3.tar.gz`
- `gcc-g++-3.4.5-20060117-3.tar.gz`
- `gcc-g77-3.4.5-20060117-3.tar.gz`
- `gcc-objc-3.4.5-20060117-3.tar.gz`
- `gdb-6.8-mingw-3.tar.bz2`
- `gdbm-1.8.3-MSYS-1.0.11-1.tar.bz2`
- `gettext-0.16.1-1-bin.tar.bz2`
- `gettext-0.16.1-1-dll.tar.bz2`
- `gettext-0.16.1-MSYS-1.0.11-1.tar.bz2`

- gettext-devel-0.16.1-MSYS-1.0.11-1.tar.bz2
- inetutils-1.3.2-40-MSYS-1.0.11-2-bin.tar.gz
- libiconv-1.11-1-bin.tar.bz2
- libiconv-1.11-1-dll.tar.bz2
- libiconv-1.11-MSYS-1.0.11-1.tar.bz2
- libtool1.5-1.5.25a-1-bin.tar.bz2
- libtool1.5-1.5.25a-1-dll.tar.bz2
- libtool1.5-1.5.25a-20070701-MSYS-1.0.11-1.tar.bz2
- lndir-6.8.1.0-MSYS-1.0.11-1.tar.bz2
- lpr-1.0.1-MSYS.tar.gz
- lzma-4.43-MSYS-1.0.11-1-bin.tar.gz
- make-3.81-MSYS-1.0.11-2.tar.bz2
- mingw-runtime-3.14.tar.gz
- mingw-utils-0.3.tar.gz
- minires-1.01-1-MSYS-1.0.11-1.tar.bz2
- MSYS-1.0.11-20071204.tar.bz2
- msysCORE-1.0.11-2007.01.19-1.tar.bz2
- openssh-4.7p1-MSYS-1.0.11-1-bin.tar.gz
- openssl-0.9.8g-1-MSYS-1.0.11-2-bin.tar.gz
- openssl-0.9.8g-1-MSYS-1.0.11-2-dev.tar.gz
- openssl-0.9.8g-1-MSYS-1.0.11-2-dll098.tar.gz
- perl-5.6.1-MSYS-1.0.11-1.tar.bz2
- perl-man-5.6.1-MSYS-1.0.11-1.tar.bz2
- regex-0.12-MSYS-1.0.11-1.tar.bz2
- tar-1.19.90-MSYS-1.0.11-1-bin.tar.gz
- texinfo-4.11-MSYS-1.0.11-1.tar.bz2
- vim-7.1-MSYS-1.0.11-1-bin.tar.gz
- w32api-3.11.tar.gz
- zlib-1.2.3-MSYS-1.0.11-1.tar.bz2

This file list might contain file which are not absolutely mandatory for Liquid War 6, for instance the Fortran 77 compiler is absolutely useless, but installing it won't harm either. Some packages might unzip things the right way, but some do it in a subfolder. You might need to run commands like:

```
cp -r coreutils*/* .
rm -rf coreutils*
```

Get rid of useless files:

```
rm ._.DS_Store .DS_Store
```

It's also mandatory to move everything that has been installed in `/usr` or `/usr/local` to `/` since MSYS has some builtin wizardry which maps `/usr` on `/`. You need to do this if you don't unzip files from a MinGW shell, which is obviously the case when you first install it. Usefull command can be:

```
mv usr/* .
rmdir usr
```

Next, `libintl` is not correctly handled/detected by LW6, and can raise an error like `"gcc.exe: C:/msys/local/lib/.libs/libintl.dll.a: No such file or directory"` so one needs to copy some libraries in `/usr/local/lib/.libs/`:

```
mkdir local/lib/.libs
cp local/lib/libintl.* local/lib/.libs/
```

Another step is to edit `/etc/profile` and add lines like:

```
export CFLAGS="-g -I/usr/local/include"
export LDFLAGS="-L/usr/local/lib"
export GUILE_LOAD_PATH="C:\\MSYS\\local\\share\\guile\\1.8\\"
```

Close and re-launch your msys shell (rxvt) so that these changes take effect. Check that those values are correctly set:

```
env | grep FLAGS
env | grep GUILE
```

Finally, your MSYS environment is (hopefully...) working.

Now you need to compile the following programs, from source. Files are [mirrored on ufoot.org](#) for your convenience, however these might be outdated. Still, there are known to work. Proceed like if you were under a POSIX system. Some packages use the `--disable-rpath` switch, there are various reasons for which [rpath is an issue](#). In the same manner, `--disable-nls` when linking against `libintl` or `libiconv` was painful.

- [pthreads-win32](#), `untar pthreads-w32-2-8-0-release.tar.gz` then `make clean GC`; `cp pthread.h sched.h /usr/local/include/`; `cp pthreadGC2.dll /usr/local/bin/`; `cp libpthreadGC2.a /usr/local/lib/`
- [GNU MP](#), `untar gmp-4.2.2.tar.gz` then `./configure && make && make install`
- [Guile](#), `untar guile-1.8.5.tar.gz`. Edit `libguile/guile.c` and insert `#undef SCM_IMPORT` just before `#include <libguile.h>`. Edit `./libguile/threads.c` and place `struct timespec { long tv_sec; long tv_nsec; };` just before `#include "libguile/_scm.h"`. Then `./configure --disable-nls --disable-rpath --disable-error-on-warning --without-threads && make && make install`. The `GUILE_LOAD_PATH` value must be correctly set for `guile-config` to work. For unknown reasons, running `guile` can throw a stack overflow error. Don't panic. See [bug 2007506 on SourceForge.net](#) for an explanation on why the Guile binary shipped with MSYS is not suitable for Liquid War 6.
- [expat](#), `untar expat-2.0.1.tar.gz` then `./configure && make && make install`
- [SQLite](#), `untar sqlite-amalgamation-3.5.9.tar.gz` then `./configure && make && make install`
- [libpng](#), `untar libpng-1.2.29.tar.gz` then `./configure && make && make install`
- [libjpeg](#), `untar jpegsrc.v6b.tar.gz` then `./configure && make && make install && make install-lib`
- [libcURL](#), `untar curl-7.18.1.tar.gz` then `./configure --without-ssl && make && make install`
- [FreeType 2](#), `untar freetype-2.3.5.tar.gz` then `./configure && make && make install`

- **libogg**, `untar libogg-1.1.3.tar.gz` then `./configure && make && make install`
- **libvorbis**, `untar libvorbis-1.2.0.tar.gz` then `LDFLAGS="$LDFLAGS -logg" && ./configure && make && make install`
- **SDL**, `untar SDL-1.2.13.tar.gz` then `./configure && make && make install`
- **SDL_image**, `untar SDL_image-1.2.6.tar.gz` then `./configure && make && make install`
- **SDL_mixer**, `untar SDL_mixer-1.2.8.tar.gz` then `./configure && make && make install`
- **SDL_ttf**, `untar SDL_ttf-2.0.9.tar.gz` then `./configure && make && make install`

For your convenience, a zip file containing a complete MSYS "Liquid War 6 ready" environment is available. It is simply the result of all the operations described above. Simply unzip **msys-for-liquidwar6-20080819.zip** (about 240 megs) in `C:\MSYS\`. All dependencies compiled in `/local` have been generated using the command:

```
cd /usr/local/src
./msys-for-liquidwar6-build.sh > ./msys-for-liquidwar6-build.log 2>&1
```

Note that this script doesn't do everything, you'll still need to edit Guile source code and patch it manually.

It might even be possible to use this MSYS environment under **Wine**. Simply unzip it under `$HOME/.wine/drive_c`, and run `wine "$HOME/.wine/drive_c/windows/system32/cmd.exe" /c "c:\msys\msys.bat"` and with luck, you'll get a working shell. Note that this might allow you to compile the game, but running it is another story. Consider this MSYS over Wine trick as a hack enabling the use of free software only when compiling for Microsoft proprietary platform. It is not a reasonable way to run the game. If running under a UNIXish platform, or better, GNU, simply run native code. Use the Windows 32-bit port only if you are jailed on a Microsoft system.

Now, let's come to the real meat, untar the Liquid War 6 source tarball, launch your MSYS shell, and:

```
./configure
make
make install
```

Now the binary is in `src/.libs/liquidwar6.exe` (beware, `src/liquidwar6.exe` is only a wrapper). This binary is an MSYS/MinGW binary, so it reads paths "la" Microsoft, that is, it has no knowledge of what `/usr` is, for instance. It requires paths starting by `C:\`.

3.7.3 Mac OS X port

This is still experimental. Basically, install MacPorts, and most dependencies with, except for SDL which you compile from source. The idea is to compile SDL using the native OS X bindings (and not some other GL files you could have in `/opt/local` installed by MacPorts), then compile the game and other SDL dependencies against this SDL.

The `SDL_mixer` library might need to be told to compile itself without dynamic ogg support. By default it seems that it tries to load `libvorbisfile.dylib` at runtime, and it can fail. To disable this dynamic loading, use for instance :

```
/configure --prefix=/opt/extra --enable-music-ogg --disable-music-ogg-shared
```

Also, it might seem obvious for Mac OS X users, but there are some important issues related to compiling options and handling dynamic libraries at runtime.

- The command `ldd` does not exist, run `otool -L` instead.
- The equivalent of `LD_LIBRARY_PATH` is `DYLD_LIBRARY_PATH`.
- The extension for shared binaries is `.dylib` and not `.so`.
- You might need to set the `OBJCFLAGS` environment variable along with `CFLAGS` because the Mac OS X port uses some Objective-C code.

It is very important to have the right SDL flags when linking the Liquid War 6 binaries. For instance it could be:

```
-I/opt/extra/include -I/opt/local/include -Wl,-framework -Wl,CoreFoundation -I/opt/local/in
```

The point is to have Cocoa and OpenGL support. Depending on the way you installed SDL, you might also need to include an SDL framework support, this is mostly if you installed SDL from .dmg binary images, and not from source with the command line. A typical output of `sdl-config --libs` is:

```
-L/opt/extra/lib -lSDLmain -lSDL -Wl,-framework,Cocoa
```

Another important issue is to include `SDL.h`, which in turn includes `SDLmain.h`, in all the .c source files defining the standard `main` function. This is done in `liquidwar6` but should you try to link yourself on `liquidwar6` libraries and/or hack code, you must do this or you'll get errors when running the game. Such errors look like:

```
*** _NSAutoreleaseNoPool(): Object 0x420c90 of class NSCFNumber autoreleased with no pool i
```

The reason is that SDL replaces your `main` with its own version of it. One strong implication is that all the dynamic loading of SDL, which works on standard GNU/Linux boxes, won't work under Mac OS X, since SDL hard codes itself by patching `main` with `#define` C-preprocessor commands.

A .dmg file (disk image) containing a `Liquid War 6.app` folder (OS X application) is available for your convenience. It might work or not. In doubt, compile from source. The complicated part about this package (a "bundle" is OS X language) is that it needs to embed various libraries which are typically installed in `/opt` by MacPorts on a developer's machine. So to build this package a heavy use of the utility `install_name_tool` is required, normally all libraries needed ship with the binary, remaining dependencies concern frameworks which should be present on any working Mac OS X install. Still, this is only theory. Needs to be widely tested.

The layout of the bundle follows:

- `./Contents/Info.plist`: metadata, bundle description file
- `./Contents/MacOS`: contains the main binary `liquidwar6` as well as all specific low-level libraries
- `./Contents/Resources/data`: general game data, does not contain maps.
- `./Contents/Resources/music`: music for the game.
- `./Contents/Resources/map`: system maps, you can put your own maps (or "extra" maps) here if you want all users to share them.
- `./Contents/Resources/script`: Liquid War 6 specific scripts, the scheme scripted part of the program.

- `./Contents/Resources/guile`: common shared Guile scripts, part of Guile distribution.
- `./Contents/Resources/doc`: documentation in HTML and PDF formats.

Additionally, the Mac OS X port uses `/Users/<username>/Library/Application Support/Liquid War 6/` to store configuration file, logs, etc. It does not use `$HOME/.liquidwar6` like the default UNIX port.

The Mac OS X port also has a special behavior, in order to load some libraries with `dlopen` (SDL_image does this with `libpng` and `libjpeg`) it needs to set `DYLD_FALLBACK_LIBRARY_PATH` to a value that contains these libraries. This is typically in the bundle distributed on the disk image. On a developer's computer this problem does not appear for those libs are often in places like:

- `/usr/local/lib`
- `/usr/X11/lib`
- `/opt/local/lib`

So the program sets `DYLD_FALLBACK_LIBRARY_PATH` (but not `DYLD_LIBRARY_PATH` else it breaks internal OS X stuff which relies, for instance, on `libjpeg` library that has the same name but different contents) but it needs to do it before it is even run, else the various `dyld` functions do not acknowledge the change. That is, calling the C function `setenv()`, even before `dlopen()`, has no effect. So the program calls `exec()` to re-run itself with the right environment variable. This is why, on Mac OS X, there are two lines (exactly the same ones) displaying the program description when it is started in a terminal.

3.7.4 GP2X port

This is not working yet, but there are good hopes that some day, Liquid War 6 can run on a **GP2X console**. This handheld gaming device uses a **Linux kernel** on an ARM processor, does support most **GNU** packages through cross-compilation, and has basic **SDL** support.

To compile Liquid War 6 for GP2X, you first need to set up a working “toolchain”. It's suggested you do this on a working GNU/Linux box. There are several solutions, the recommended one being **Open2x**. Read the online documentation on **how to install Open2x**.

Basically, the following should work:

```
mkdir /opt/open2x # check that you can write here
svn co https://open2x.svn.sourceforge.net/svnroot/open2x/trunk/toolchain-new open2x-toolchain
./open2x-gp2x-apps.sh
cd open2x-toolchain
```

Then, for your environment to be complete, you need to set up some environment variables. For instance:

```
export OPEN2X_SYSTEM_PREFIX=/opt/open2x/gcc-4.1.1-glibc-2.3.6/arm-open2x-linux
export GP2X_USER_PREFIX=$HOME/gp2x
```

```
export CC=${OPEN2X_SYSTEM_PREFIX}/bin/arm-open2x-linux-gcc
export CPP=${OPEN2X_SYSTEM_PREFIX}/bin/arm-open2x-linux-cpp
export CXX=${OPEN2X_SYSTEM_PREFIX}/bin/arm-open2x-linux-g++
export AS=${OPEN2X_SYSTEM_PREFIX}/bin/arm-open2x-linux-as
```



```
export CFLAGS=''-I${OPEN2X_PREFIX}/include -I${GP2X_USER_PREFIX}/include''
export CPPFLAGS=''-I${OPEN2X_PREFIX}/include -I${GP2X_USER_PREFIX}/include''
export CXXFLAGS=''-I${OPEN2X_PREFIX}/include -I${GP2X_USER_PREFIX}/include''
export LDFLAGS=''-L${OPEN2X_PREFIX}/lib -L${GP2X_USER_PREFIX}/lib''
```

```
export PATH='''${GP2X_USER_PREFIX}/bin:${OPEN2X_SYSTEM_PREFIX}/bin:$PATH'''
```

In this setting, there's a user `$HOME/gp2x` directory which will contain all the Liquid War 6 related libraries while the `/opt/open2x` remains untouched.

Then you need to install the requirements. All these packages need to be cross-compiled. To make things clear and non-ambiguous, even if you have `CC` set up in your environment variables, pass `--build` and `--host` arguments to the `./configure` script of all these packages. A typical command is:

```
./configure --build=i686-pc-linux-gnu --host=arm-open2x-linux --prefix=${GP2X_USER_PREFIX}
```

Here's the list of the requirements:

- **libtool**. Install a 1.5.x version, and not 2.2.x. For some reasons, 2.2.x is unusable, it keeps on complaining that `lt_libltdl_LTX_preloaded_symbols is not defined`. Apparently the bug has already been [reported](#). For Liquid War 6 needs, 1.5.x is very fine, use 1.5.26 for instance, it works well.
- **GNU MP** (version 4.3.1 reported to work)
- **Guile** (version 1.8.7 reported to work). You need to pass the `--without-threads` switch to the `./configure` script else it will try (and fail!) to run a test program. Liquid War 6 does not use Guile threads, it does use threads but uses them “directly” outside the Guile layer.
- **zlib** (version 1.2.3 reported to work). Do not use `--build` and `--host` for this one, they are unsupported. Package compiles fine anyway.
- **expat** (version 2.0.1 reported to work).
- **libpng** (version 1.2.3 reported to work).
- **libjpeg** (version 7 reported to work).
- **SQLite 3** (version 3.6.18 reported to work).
- **libcurl** (version 7.19.6 reported to work).

Next, one needs to install a special version of SDL, which targets the GP2X specifically. This is not a generic SDL, and it does have limitations, which are related to the GP2X peculiar hardware. There are [installation instructions](#) about how to do this. The following should work:

```
cvs -d :pserver:anonymous@cvs.sourceforge.net:/cvsroot/open2x login # blank password
cvs -d :pserver:anonymous@cvs.sourceforge.net:/cvsroot/open2x co libs-gp2x
```

3.8 Coding guidelines

3.8.1 Project goals reminder

One of the purposes of Liquid War 6 is to make a cleaner implementation of Liquid War than the previous one, namely [Liquid War 5](#). While the latter has achieved the practical goal of providing a playable implementation of the game, it failed at providing an evolutive

platform. Network capabilities were finally added to **Liquid War 5**, but anyone who played on Internet with someone a few hundreds of milliseconds away would agree that it's far from being perfect. The main reason for this is that it is really hard to hack on **Liquid War 5**, especially when you are not the core developer. The core developer himself, even knowing all the various hacks in the game, is very quickly lost when trying to implement major changes.

To put it short, **Liquid War 5** is a global variable hell, a pile of hacks on top of a quick and dirty implementation. Still, it works.

With Liquid War 6, the idea is to take the time to make something stable, something nice which will enable developers to implement the cool features, and have fun along the way. Of course, this is only a dream, and in the (hopefully "very") long run, Liquid War 6 will also end up as a big unmaintainable mess, like any real-life program, until then, it should remain hackable.

3.8.2 Common sense

Here are a few guidelines which I think are common sense advice, but they are still worth mentioning:

- try and respect the **GNU coding standards**;
- absolutely no **strcpy** or **sprintf** anywhere in the code. Nowhere. Use their equivalent **strncpy** and **snprintf** systematically, as they are part of the glibc and are an order of magnitude safer. Moreover, Liquid War 6 provides wrappers, such as **lw6sys_new_sprintf** which handles all the nasty dirty memory allocation stuff for you;
- keep global variables for when there is something truly global, and even in that case try to fit them in clearly identified structures.

3.8.3 Unitary tests

Each of the internal libraries in Liquid War has a "test" program associated with it. For instance **liquidwar6sys-test** is associated to **libliquidwar6sys**, and its purpose is to test the features of this library.

While it is fairly easy to test out unitary functions which require no peculiar context, testing high-level functions which requires files, graphical and possibly network contexts to exist is obviously harder to achieve. There's no easy way to draw the line, but the idea is to put in these test executables as many features as possible, to be sure that what is tested in them is rock solid, bullet proof, and that one can safely rely on it and trust that code when running it in a more complex environment.

These test executables are also very good places to see a library API in action, find code fragments, and make experiments.

3.8.4 Memory allocation

Liquid War 6 provides macros to allocate and free memory. One should use them systematically, except when trying to free something allocated by another library, and in very special cases, mostly concerning low-level operations which are seldom hacked on.

Usage of macros **LW6SYS_MALLOC**, **LW6SYS_CALLOC** and **LW6SYS_FREE** is straightforward, read any random chunk of code, for instance **./src/lib/sys/sys-test.c** to see them in action. They are defined in **sys/sys.h**.

Once used, these macros will track every single call to `malloc` and `free`, and if there's a difference, it will report it. It will also help you by showing what's in the non-freed memory area, at which line of code it has been allocated, and when. This is very useful to track down memory leaks. Of course a debugger could tell you some of these informations, but experience shows that when you encounter a memory bug, it's very often impossible to reproduce it. So you waste time trying to reproduce the bug, whereas with this tool you have the information reported just when the problem happens.

3.8.5 Private and public interfaces

Each library exports a public interface and hides its internal. Since Liquid War 6 uses standard C and no C++, there's no real standard way to handle public/private features. The convention used in Liquid War 6 is to show internal structures as opaque pointers (`void *`) whenever some function needs to operate on a structure which has possibly private fields. This way the caller function has no way to access the internals, and we are sure that no reference to any internal implementation specific feature will appear.

Here's a code excerpt from `src/gfx/setup.c`:

```
void _lw6gfx_quit(_LW6GFX_CONTEXT *context) {
    /*
     * Implementation here.
     */
    [...]
}

void lw6gfx_quit(void *context) {
    _lw6gfx_quit((_LW6GFX_CONTEXT *) context);
}
```

The function `_lw6gfx_quit` (note the “_”) is internal, declared in `internal.h` whereas the function `lw6gfx_quit` is public, and is therefore exported in `gfx.h`.

This way, functions in the program using `lw6gfx_quit` do not know what is in the `_LW6GFX_CONTEXT` structure, and they need not know it.

This does not mean it is not possible to have public structures, only these structures must reflect some truly public, accessible and safe to access structures.

3.8.6 Audit the code

Liquid War 6 is regularly audited with automated tools. You might need to pass `--enable-gcov` to `--configure` if you want to use these tools yourself. More precisely:

- `lcov` is run, ideally with each release - but it's not guaranteed, check the output date and time - and the output is available online on <http://www.gnu.org/software/liquidwar6/coverage/>.
- `GNU global` is run too, ideally with each release - again, check the output date and time - and the output is available online on <http://www.gnu.org/software/liquidwar6/global/>.
- `pmccabe` reports cyclomatic complexity. It shows where the code is too complex and should probably be rewritten. Output is post-processed using `pmccabe2html` from `gnulib`. The output is available online on <http://www.gnu.org/software/liquidwar6/cyclo/>.

- **Valgrind** is run as well, it should report absolutely no leak on all the core sub-libraries, eg running `liquidwar6ker-test` or `liquidwar6ker-pil`. Bits of code which depend on other libraries are a different story, for some projects on which Liquid War 6 depends might, for some reason, raise warnings. But as far as Liquid War 6 is concerned, the goal is simple: zero leak.

Those tools certainly don't guarantee the code is perfect, but they do help improving the quality of the program. If you hack, it's recommended you give them a try.

3.9 Using the console

The console can be activated by passing `--display-console` when starting the game or by using the system options menu.

When the console is activated, a `lw6>` prompt should appear in the terminal which launched the program. If you started Liquid War 6 by clicking on an icon, console probably won't work at all since `stdout` and `stdin` won't be attached to anything.

The console allows you to type arbitrary Scheme/Guile code.

Try, for instance:

```
(+ 1 2)
(display "foo\n")
```

You can really break things with this console, it gives you a direct access to all the program internals. At least, all the values which are accessible through the script interface, that is, many of them.

You can call any internal C function which has been exported to Guile, here are some examples:

```
(c-lw6sys-timestamp)
(c-lw6bot-get-backends)
(c-lw6sys-sleep 2.0)
(lw6-config-get-number "zoom")
(lw6-config-set-number! "zoom" 0.9)
(lw6-config-get-number "zoom")
```

While syntax (and possibly other) errors will be trapped by the interpreter, note that if you break things inside the game by, say, changing some global value, or in a general manner cause an error elsewhere in the code, the game will really raise a fatal error and stop. That's how you can "break things".

Still, this console is a very powerful tool, very useful for debugging but also for tweaking the game without restarting it and/or navigating through the menu interface.

3.10 Advanced tweaking

3.10.1 Hacking resources

Liquid War 6 tries to have as few hardcoded data as possible. So many constants, and pretty much all the image files, are accessible in the data directory. You can know where it is by launching `liquidwar6 --show-data-dir`. If you look in this directory you'll find different files, among them XML files.

Let's take an example. Try and find the file `gfx/gl/hud/floating/gl-floating-const.xml`. Edit the line with the `clock-y1` entry. Change the number after `"value"`. Re-run the program. Play a game. What happens? Logically you should see that "something" is not displayed at the same place than before.

You could also modify the textures (JPEG and PNG files). In a general manner it's more cautious to keep them the same size but it depends, sometimes other sizes will work as well.

Many of these parameters are really too technical and obscure to have their place in the main config file (which is already rather big). Use at your own risks, you can really break things touching this, but you can also find out lots of things can be tuned.

3.10.2 Optimize for speed

Todo...

3.11 Writing modules

Todo...

3.12 Use as a library

Todo...

3.13 Network protocol

This section describes how Liquid War 6 handles network messages. Note that for now this is purely theoretical, more of a draft, a plan, it might change before being implemented.

3.13.1 No server, no client, only nodes

Liquid War 6 does not really have the notion of server or client, any instance of the program can act as both server and client, therefore we use the term node.

A node listens on a given port, in both TCP and UDP, and can connect to other nodes on the same port. The main identifier of a node is its public url, which is typically of the form `http://<ip-address>:<port>/`. This url is very important for it is (or at least should be) a unique identifier of the node over the network.

Liquid War has 3 ways to communicate:

- raw TCP, this is the LW6 protocol, the easiest to implement and debug, probably the most reliable one, but not always the fastest. This involves the two modules `mod-tcp` and `mod-tcpd`.
- HTTP over TCP, this is a hack which allows the game to communicate through HTTP proxies for instance. Additionally, the fact any node is a web server enables peering with a simple web browser. Web server facility requires `mod-httpd` and client part requires `mod-http` which might or not be available, depending on how the game was compiled.
- raw UDP, this is another version of the LW6 protocol, this is in theory the fastest way to communicate, it requires `mod-udp` and `mod-udp` to work. Using UDP only was not an option when conceiving Liquid War since it's interesting to have other solutions if, for instance, a firewall does not allow you to use UDP the way you want.

On each of these channels, messages can be exchanged in two modes, an “out of band” mode (AKA “oob”), and a regular message oriented, here we speak of “connection”.

3.13.2 Out of band messages

There are only 3 out of band messages:

- **PING**: requests for a simple PONG `http://server:port/` answer, this is just to check if a server is a valid server, and if the URL we used to connect on it is the correct one.
- **INFO**: requests for a list of key/attributes pairs, which describe the node, telling for instance its version, its uptime, and so on.
- **LIST**: requests for a list of other nodes this node is aware of.

You can test out of band messages by simply connecting on your server with a command like:

```
telnet localhost 8056
```

At the telnet prompt, simply type:

```
INFO
```

and press return, and you should have a description of your node.

The complete syntax of OOB messages is:

```
<COMMAND> [password] [url]
```

The `password` and `url` parameters are optional. `password` can contain either the plain text password or a checksum calculated from the password which is, for security reasons, seeded with the public url of the node we're connecting to, so that this password can't be re-used on another node. `url` is simply a clue we give to the other node to help find us, the other node will automatically try to detect our IP address and use standard LW6 port 8056, but if for some reason we use a different setting, this is a good way to pass the hint.

As far as OOB is concerned, TCP and UDP work almost the same, HTTP is a bit different, the OOB commands are accessed through the following URLs:

- `/ping.txt`
- `/info.txt`
- `/list.txt`

3.13.3 Other raw technical stuff (WIP)

TCP messages:

```
LW6 [<passwd>] <version> <client-id>
<from-id> <to-id> <serial> <i> <n> <sig> MSG1
<from-id> <to-id> <serial> <i> <n> <sig> MSG2
```

TCP oobs:

```
<return> # only works anonymous, same as INFO
INFO [<passwd>] [<public-url>]
LIST [<passwd>] [<public-url>]
PING [<passwd>] [<public-url>]
```

UDP messages:

```
LW6 [<passwd>] <version> <client-id> <from-id> <to-id> <serial> <i> <n> <sig> MSG1
LW6 [<passwd>] <version> <client-id> <from-id> <to-id> <serial> <i> <n> <sig> MSG2
```

UDP oobs:

```
INFO [<passwd>] [<public-url>]
LIST [<passwd>] [<public-url>]
PING [<passwd>] [<public-url>]
```

HTTP messages:

client id & password passed in HTTP headers

```
/lw6/version/<from-id>/<to-id>/<serial>/<i>/<n>/sig/MSG1
/lw6/version/<from-id>/<to-id>/<serial>/<i>/<n>/sig/MSG2
```

HTTP public URLs:

```
/ -> index.html
/index.html
/favicon.ico
/screenshot.jpeg
/robots.txt
/gpl.txt
/info.txt
/list.txt
/ping.txt
```

MSG syntax:

```
<round> <server-id> <command> <arg1> ... <argN>
```

COMMAND examples:

```
2 1234abcd1234abcd REGISTER
3 1234abcd1234abcd ADD 5678 YELLOW
4 1234abcd1234abcd SET 5678 20 5
10 1234abcd1234abcd NOP
400 1234abcd1234abcd REMOVE 5678
1000 1234abcd1234abcd UNREGISTER
```

Sig is a checksum on string:

```
<from-id> <to-id> <serial> <i> <n> MSG
```

3.14 Using GNU Arch

3.14.1 About GNU Arch

Since March, 4th 2010, Liquid War 6 uses **GIT** to handle source code, track changes, branches, and the rest. It replaces the **GNU Arch** repository. This old repository contains all sources up to version 0.0.7beta, following versions, including 0.0.8beta, must be retrieved from GIT.

So the following informations only concern those who are interested in previous versions of the game. Anybody else - probably you - should use GIT instead.

See [Section 3.15 \[Using GIT\]](#), page 42.

Still, this quick Arch survival guide is kept in the documentation.

Read the [GNU Arch tutorial](#) to learn how Arch works. Note that there are many other source control managers available, some of which provide functionalities similar to GNU Arch / tla. GNU Arch has been chosen for Liquid War 6 because:

- it is Free Software,
- it is not limited to per-file commits like CVS, and supports atomic commits involving several files,
- it is distributed,
- it enables developers to sign each of their contributions,
- it was already available back in 2005.

3.14.2 Getting the latest version from the repository

The repository for Liquid War 6 is accessible on <http://arch.savannah.gnu.org/archives/liquidwar6>. This is a read-only access, but with the distributed nature of GNU Arch, it still allows you to keep track of your own changes, and submit patches. Accessing it in read/write mode with sftp requires a Savannah account and special rights on the Liquid War 6 project.

Here are typical commands one can use to get Liquid War 6 source from the GNU Arch repository:

```
tla register-archive http://arch.savannah.gnu.org/archives/liquidwar6
tla get -A liquidwar6@sv.gnu.org liquidwar6--beta
```

All the patches in the archive are signed with [GnuPG](#), so you can check their authenticity with [my public key](#).

You might need to edit your `$HOME/.arch-params/signing/=default.check` file and put the following text in it:

```
tla-gpg-check gpg_command="gpg --verify-files -"
```

3.14.3 Setting up your own arch repository

This section is for those who want to hack the game and set up their own repositories. This will enable you to keep track of your patches, package them, and help the core maintainer merging them in the main repository.

You can introduce yourself and create a repository by issuing commands like:

You can introduce yourself and create a repository by issuing commands like:

```
tla my-id me@home.net
tla register-archive me@home.net--2008 /home/me/tla-archives
```

Then, you can get create your own repository, with a command like:

```
tla tag -S liquidwar6@sv.gnu.org/liquidwar6--beta--0.1 me@home.net--2008/liquidwar6--b
```

The idea is that you create, locally, a depot which has a name that matches the name on [savannah](#) (this is for convenience, you could technically give it any name...) and indicate that they represent, today, the same thing.

You can get a working copy of your depot with the command:

```
tla get me@home.net--2008/liquidwar6--beta--0.4
```

This will create a complete source tree, which you are free to modify, this is where you should hack.

3.14.4 Synchronizing your repository with upstream releases

To synchronize yourself with upstream developments, go into your copy (the directory created by `tla get`) and type:

```
tla star-merge liquidwar6@sv.gnu.org/liquidwar6--beta--0.1
```

This will apply locally all the changes that happened since the last synchronization. Of course this is one way to work, you can decide to cherry pick patches and such stuff, but for most daily uses, a good'ol `star-merge` is fine.

Not that `star-merge` will only apply patches on your working copy, not on your repository. The only way to actually commit the modifications on the repository is to use the `commit` command.

3.14.5 Submitting patches

When using Arch, you can of course still send patches created with `diff`, or even send updated files directly, the way you would without revision control.

But it can be more convenient to either

- Send the patches stored in the depot (`/home/me/tla-archives` in our example).
- Make patches using `tla mkpatch`.

Here's an example of an `mkpatch` command, and which will compute the differences between a previous `liquidwar6--beta--0.4--patch-2` snapshot and a not yet committed latest version:

```
tla mkpatch {arch}/++pristine-trees/unlocked/liquidwar6/liquidwar6--beta/liquidwar6--b
```

This will create a `my-patch` directory, which can be gzipped and sent by mail.

3.14.6 Recover from broken lock

Sometimes, when signing a patch, you might enter the wrong passphrase several times, or you might press CTRL+D inadvertently. In that case, `tla` will be in a half-broken state, telling you it can't acquire revision lock... A quick workaround for this is to go to the depot, find the latest patch, and in this repository, create the following folders:

```
++revision-lock/+contents
```

Both are directories, note the two `++` and the single `+`. the `+contents` directory can be empty. Once you've done this, try to re-commit.

3.15 Using GIT

3.15.1 About GIT

There's no CVS or Subversion (AKA "SVN") source depot for Liquid War 6. Instead, a **GIT** depot is used. GIT has many advantages over other source control managers (SCM), among them, it's distributed, like **GNU Arch**.

You can find interesting informations on GIT here:

- <http://git-scm.com/documentation>
- <http://savannah.gnu.org/maintenance/UsingGit>
- <http://savannah.gnu.org/git/?group=liquidwar6>

3.15.2 Getting the latest source

Simply install git and run the following command:

```
git clone git://git.sv.gnu.org/liquidwar6.git
```

If you are behind a firewall and can't have direct TCP/IP access:

```
git clone http://git.sv.gnu.org/r/liquidwar6.git
```

Additionally, source can be browsed online here: <http://git.savannah.gnu.org/cgi/liquidwar6.git>

3.15.3 Developer access

You need an ssh access on [Savannah](#) and appropriate rights on the project, then you can type:

```
git clone login@git.sv.gnu.org:/srv/git/liquidwar6.git
```

3.15.4 Submitting patches

If you have developer access to the project, then a simply `git push` will commit your changes.

If not (that is, if you checked out anonymously using `git clone git://git.sv.gnu.org/liquidwar6.git` for instance, you can still submit patches. Follow these steps:

- edit the code, make your patches, commit to your local GIT tree
- run `git format-patch -p origin` this command will generate `.patch` files, one for each of your commits, which you can send by email. They can be easily integrated in the main source tree by using `git apply <file.patch>`.

Note that you can need to run `git format-patch -p master` (with `master` instead of `origin`) if not using a fresh checkout. Also consider adding the `--stdout` switch, eg `git format-patch -p master --stdout > my-changes.patch` if you're not using a fresh checkout.

4 Reference

This chapter is a technical reference. Most of its content is self-generated by the program itself. That is to say, most of its content is already available to you if you have the game installed. Running `liquidwar6 --list` and `liquidwar6 --about=<keyword>` is very likely to give you the very same informations, the advantage being that you'll be sure the information is up-to-date and corresponds to the exact version of the program you have. However, publishing this in a reader-friendly way is convenient, plus it enables web search engines to harvest the content.

4.1 Basic options

4.1.1 about

`--about=<value>` [Command-line option]

Type: string

Will allow you to get informations about a given keyword. Let's say that, for instance, you want informations about the keyword 'map-path'. Simply run '`liquidwar6 --about=map-path`'. Note that this internal self-documentation system can describe command line switches as well as XML config file parameters or environment variables, and even some Guile script functions. The '-list' command line switch will give you the list of all available keywords.

4.1.2 audit

`--audit` [Command-line option]

Display all path values, defaults and current settings. This output is very usefull to track down problems such as missing directories, broken installations. If you get an error message that suggests some file is missing, then give this option a try.

4.1.3 copyright

`--copyright` [Command-line option]

Returns the copyright notice for the program.

4.1.4 debug

`--debug` [Command-line option]

Enables debug mode. This will turn on maximum log information, and display everything on stderr, even messages which are normally only stored in the log file.

4.1.5 defaults

`--defaults` [Command-line option]

Clears the config file and run the game with default settings. Use this if you suspect you have broken something by tweaking user settings, or when upgrading the game to a new version.

4.1.6 help

--help [Command-line option]
Returns a short help for the program.

4.1.7 host

--host [Command-line option]
Display all known system host properties, including os and cpu informations.

4.1.8 list

--list [Command-line option]
Returns the list of all keywords which can be queried for information. This includes command-line options, environment variables, and so on. This is the companion option of '-about'. Results obtained with '-list' can be passed to '-about'.

4.1.9 modules

--modules [Command-line option]
Tells which modules have been enabled when the game was compiled. It's still possible to add or remove modules afterwards, but this option allows you to know how things were at first.

4.1.10 pedigree

--pedigree [Command-line option]
Display all build values, these are general constants which can help debugging, tracing what binary you are running, and so on. It's a good idea to take a look at the output of 'pedigree' if you have problems running the game.

4.1.11 test

--test [Command-line option]
Runs a (hopefully) complete test suite which will call most internal Liquid War 6 functions and check out whether they work, in a simple context, without any game interference. Useful for troubleshooting.

4.1.12 version

--version [Command-line option]
Returns the version of the program, as defined by the GNU Coding Standards.

4.2 Doc options

4.2.1 example-hints-xml

--example-hints-xml [Command-line option]
Dumps on stdout an example hints.xml file. Such a file is normally shipped with the game. It is indeed generated using this command.

4.2.2 example-rules-xml

`--example-rules-xml` [Command-line option]
Dumps on stdout an example options.xml file. Such a file is normally shipped with the game. It is indeed generated using this command.

4.2.3 example-style-xml

`--example-style-xml` [Command-line option]
Dumps on stdout an example style.xml file. Such a file is normally shipped with the game. It is indeed generated using this command.

4.2.4 list-aliases

`--list-aliases` [Command-line option]
List the keyword aliases. These are here for convenience.

4.2.5 list-doc

`--list-doc` [Command-line option]
List documentation-related command line options. These commands allow you to list all the keywords related to a given domain.

4.2.6 list-funcs

`--list-funcs` [Command-line option]
List the C-functions which are exported to Guile, thus usable in scripts.

4.2.7 list-graphics

`--list-graphics` [Command-line option]
List graphics options (resolution, fullscreen...).

4.2.8 list-hooks

`--list-hooks` [Command-line option]
List user-modifiable hooks.

4.2.9 list-input

`--list-input` [Command-line option]
List input (AKA controls) related options. Use these to change keyboard, joystick and mouse settings.

4.2.10 list-map

`--list-map` [Command-line option]
List map-related entries, excluding rules.xml, hints.xml and style.xml entries.

4.2.11 list-map-hints

`--list-map-hints` [Command-line option]
List 'hints.xml' entries. These parameters enable you to modify the behavior of the map loader.

4.2.12 list-map-rules

`--list-map-rules` [Command-line option]
List 'options.xml' entries. These parameters enable you to modify the gameplay.

4.2.13 list-map-style

`--list-map-style` [Command-line option]
List 'style.xml' entries. These parameters enable you to modify the aspect of the game.

4.2.14 list-network

`--list-network` [Command-line option]
List network options.

4.2.15 list-path

`--list-path` [Command-line option]
List parameters which allow you to override the defaults of the game, and force the game your own file paths and directories.

4.2.16 list-players

`--list-players` [Command-line option]
List player-related entries, that is to say 'who plays'.

4.2.17 list-quick

`--list-quick` [Command-line option]
List quick help entries, this includes the GNU standard options and a few troubleshooting tools.

4.2.18 list-show

`--list-show` [Command-line option]
List command-line options which begin with '-show-...'. These will display on the console many internal parameters. Usefull when debugging.

4.2.19 list-sound

`--list-sound` [Command-line option]
List sound options (volume...).

4.2.20 list-tuning

`--list-tuning` [Command-line option]
List advanced options which can be used for fine-tuning the game.

4.3 Show options

4.3.1 show-build-bin-id

`--show-build-bin-id` [Command-line option]
Shows the internal 'bin-id' value. This value does not mean anything in itself but it's supposed to change each time you compile the game.

4.3.2 show-build-cflags

`--show-build-cflags` [Command-line option]
Shows what value you should put in 'CFLAGS' (environment variable) if you want to compile programs that use Liquid War 6 as a library, and include 'liquidwar6.h'.

4.3.3 show-build-codename

`--show-build-codename` [Command-line option]
Shows the codename associated with this version, generally the name of someone famous who is war-related (a general, an emperor...).

4.3.4 show-build-configure-args

`--show-build-configure-args` [Command-line option]
Shows the arguments that have been passed to the GNU Autoconf './configure' script when building the program. This can be very usefull if you want to know how the program has been built.

4.3.5 show-build-copyright

`--show-build-copyright` [Command-line option]
Shows a very short copyright notice.

4.3.6 show-build-datadir

`--show-build-datadir` [Command-line option]
Shows the 'datadir' value as passed to the GNU Autoconf './configure' script when compiling the program. Default is '/usr/local/share'. This is the generic, non Liquid War 6 specific data directory. Liquid War 6 related data is stored elsewhere (usually in a sub-directory) see the 'data-dir' switch for more information. 'datadir' is not 'data-dir'. That's the point.

4.3.7 show-build-date

`--show-build-date` [Command-line option]
Shows the date when the binary was compiled.

4.3.8 show-build-docdir

`--show-build-docdir` [Command-line option]
Shows the 'docdir' value as passed to the GNU Autoconf './configure' script when compiling the program. Default is '/usr/local/share/doc/liquidwar6'.

4.3.9 show-build-enable-allinone

`--show-build-enable-allinone` [Command-line option]
Shows whether the 'allinone' option has been chosen when building the game. This depends on parameters passed to './configure'.

4.3.10 show-build-enable-console

`--show-build-enable-console` [Command-line option]
Shows whether the console has been enabled when building the game. This depends on parameters passed to './configure' and also on the presence of ncurses and readline.

4.3.11 show-build-enable-fullstatic

`--show-build-enable-fullstatic` [Command-line option]
Shows whether the 'fullstatic' option has been chosen when building the game. This depends on parameters passed to './configure'.

4.3.12 show-build-enable-gcov

`--show-build-enable-gcov` [Command-line option]
Shows whether the game was built with suitable information for gcov. This depends on parameters passed to './configure'.

4.3.13 show-build-enable-gprof

`--show-build-enable-gprof` [Command-line option]
Shows whether the game was built with suitable information for gprof. This depends on parameters passed to './configure'.

4.3.14 show-build-enable-gtk

`--show-build-enable-gtk` [Command-line option]
Shows whether GTK+ support has been enabled when building the game. This depends on parameters passed to './configure' and also on the presence of GTK+ headers and libs. It uses pkg-config to detect it.

4.3.15 show-build-enable-instrument

`--show-build-enable-instrument` [Command-line option]
Shows whether the game was built with the '-finstrument-functions' GCC switch. This depends on parameters passed to './configure'.

4.3.16 show-build-enable-mod-csound

`--show-build-enable-mod-csound` [Command-line option]
Shows whether the mod-csound audio backend has been enabled when building the game. This depends on parameters passed to './configure' and also on the presence of the csound library.

4.3.17 show-build-enable-mod-gl

`--show-build-enable-mod-gl` [Command-line option]
Shows whether the mod-gl graphical backend has been enabled when building the game. This depends on parameters passed to './configure' and also on the presence of SDL and related libraries.

4.3.18 show-build-enable-mod-http

`--show-build-enable-mod-http` [Command-line option]
Shows whether the mod-http network backend has been enabled when building the game. This depends on parameters passed to './configure' and also on the presence of libCurl.

4.3.19 show-build-enable-mod-ogg

`--show-build-enable-mod-ogg` [Command-line option]
Shows whether the mod-ogg audio backend has been enabled when building the game. This depends on parameters passed to './configure' and also on the presence of SDL and related libraries.

4.3.20 show-build-enable-optimize

`--show-build-enable-optimize` [Command-line option]
Shows whether the 'optimize' option has been chosen when building the game. This depends on parameters passed to './configure'.

4.3.21 show-build-enable-paranoid

`--show-build-enable-paranoid` [Command-line option]
Shows whether the game was built with paranoid memory management. This is for debugging purposes, the default already includes some controls, with turned it's really... paranoid.

4.3.22 show-build-enable-profiler

`--show-build-enable-profiler` [Command-line option]
Shows whether the game was built with Google Profiler support. This depends on parameters passed to './configure'.

4.3.23 show-build-enable-valgrind

`--show-build-enable-valgrind` [Command-line option]
Shows whether the game was built with valgrind later use in mind. This depends on parameters passed to './configure'.

4.3.24 show-build-endianness

`--show-build-endianness` [Command-line option]
Returns the endianness. 'little' corresponds to x86-like systems, 'big' to ppc-like systems.

4.3.25 show-build-gcc-version

`--show-build-gcc-version` [Command-line option]
Returns the version of the GNU C compiler which was used to compile the program.

4.3.26 show-build-gp2x

`--show-build-gp2x` [Command-line option]
Returns 1 (true) if host is a GP2X, 0 (false) if not.

4.3.27 show-build-host-cpu

`--show-build-host-cpu` [Command-line option]
Shows the host CPU, as defined by 'host_cpu' in GNU Autoconf.

4.3.28 show-build-host-os

`--show-build-host-os` [Command-line option]
Shows the host OS, as defined by 'host_os' in GNU Autoconf.

4.3.29 show-build-hostname

`--show-build-hostname` [Command-line option]
Shows the name of the host where the binary was compiled.

4.3.30 show-build-includedir

`--show-build-includedir` [Command-line option]
Shows the 'includedir' value as passed to the GNU Autoconf './configure' script when compiling the program. Default is '/usr/local/include'.

4.3.31 show-build-ldflags

`--show-build-ldflags` [Command-line option]
Shows what value you should put in 'LDFLAGS' (environment variable) if you want to link programs against libliquidwar6.

4.3.32 show-build-libdir

--show-build-libdir [Command-line option]
Shows the 'libdir' value as passed to the GNU Autoconf './configure' script when compiling the program. Default is '/usr/local/lib'. This is the generic, non Liquid War 6 specific library directory. Dedicated Liquid War 6 modules are stored elsewhere (usually in a sub-directory) see the 'mod-dir' switch for more information.

4.3.33 show-build-license

--show-build-license [Command-line option]
Shows the license of the program (GNU GPL v3 or later).

4.3.34 show-build-localedir

--show-build-localedir [Command-line option]
Shows the 'localedir' value as passed to the GNU Autoconf './configure' script when compiling the program. Default is '/usr/local/share/locale'.

4.3.35 show-build-mac-os-x

--show-build-mac-os-x [Command-line option]
Returns 1 (true) if host OS is Mac OS X, 0 (false) if not.

4.3.36 show-build-md5sum

--show-build-md5sum [Command-line option]
Shows the MD5 checksum, which has been calculated from the C source files. Complementary with 'show-build-stamp'.

4.3.37 show-build-ms-windows

--show-build-ms-windows [Command-line option]
Returns 1 (true) if host OS is Microsoft Windows 32-bit platform, 0 (false) if not.

4.3.38 show-build-package-name

--show-build-package-name [Command-line option]
Shows the package name, that is, 'Liquid War 6'.

4.3.39 show-build-package-string

--show-build-package-string [Command-line option]
Shows the package string, that is, 'Liquid War 6 <version>'.

4.3.40 show-build-package-tarname

--show-build-package-tarname [Command-line option]
Shows the package tarname, that is, liquidwar6.

4.3.41 show-build-pointer-size

--show-build-pointer-size [Command-line option]
Returns the pointer size, in bytes. Should be 4 on 32-bit systems and 8 on 64-bit systems.

4.3.42 show-build-prefix

--show-build-prefix [Command-line option]
Shows the 'prefix' value as passed to the GNU Autoconf './configure' script when compiling the program. Default is '/usr/local'.

4.3.43 show-build-stamp

--show-build-stamp [Command-line option]
Shows the build stamp. A very usefull value, more precise than the version to track down binaries. It is incremented each time the core C code is updated. It won't reflect all the programs for it does not take scripts in account, but if you are running a work-in-progress version, it might be very convenient to use this to know what your are running exactly.

4.3.44 show-build-time

--show-build-time [Command-line option]
Shows the time when the binary was compiled.

4.3.45 show-build-top-srcdir

--show-build-top-srcdir [Command-line option]
Shows the top source directory on the machine where the binary was compiled.

4.3.46 show-build-url

--show-build-url [Command-line option]
Shows the URL of the program, its homepage.

4.3.47 show-build-version

--show-build-version [Command-line option]
Shows the version. Note that this is different from the standard GNU 'version' command line option which shows a complete message with a short copyright notice. This one will just return the version, without the package tarname or anything else.

4.3.48 show-build-x86

--show-build-x86 [Command-line option]
Tells wether the CPU belongs to x86 family.

4.3.49 show-config-file

--show-config-file [Command-line option]
Shows the config file path. Default is '\$HOME/.liquidwar6/config.xml'.

4.3.50 show-cwd

`--show-cwd` [Command-line option]
Shows the current working directory, the value that the `pwd` command would return.

4.3.51 show-data-dir

`--show-data-dir` [Command-line option]
Shows the data directory path. This is where the game searches for most of its data, the most important exception being maps, which are stored elsewhere. Default is `'/usr/local/share/liquidwar6-<version>/data'`.

4.3.52 show-default-config-file

`--show-default-config-file` [Command-line option]
Shows the default config file path. Default is `'$HOME/.liquidwar6/config.xml'`.

4.3.53 show-default-data-dir

`--show-default-data-dir` [Command-line option]
Shows the default data directory path. This is where the game searches for most of its data, the most important exception being maps, which are stored elsewhere. Default is `'/usr/local/share/liquidwar6-<version>/data'`.

4.3.54 show-default-log-file

`--show-default-log-file` [Command-line option]
Shows the default log file path. Default is `'$HOME/.liquidwar6/log.csv'`.

4.3.55 show-default-map-dir

`--show-default-map-dir` [Command-line option]
Shows the default map directory. This is where builtin maps are stored. Default is `'/usr/local/share/liquidwar6-<version>/map'`.

4.3.56 show-default-map-path

`--show-default-map-path` [Command-line option]
Shows the default map search path. This is where the game searches for maps. It's the combination of command-line arguments and builtin paths. Might return more directories than the one specified in a single `'map-path=dir1:dir2'` argument.

4.3.57 show-default-mod-dir

`--show-default-mod-dir` [Command-line option]
Shows the default module directory path. This is where all dynamically loaded modules are stored. Default is `'/usr/local/lib/liquidwar6-<version>'`.

4.3.58 show-default-music-dir

`--show-default-music-dir` [Command-line option]
Shows the default music directory. This is where builtin musics are stored. Default is `'/usr/local/share/liquidwar6-<version>/music'`.

4.3.59 show-default-music-path

`--show-default-music-path` [Command-line option]
Shows the default music search path. This is where the game searches for musics. It's the combination of command-line arguments and builtin paths. Might return more directories than the one specified in a single `'music-path=dir1:dir2'` argument.

4.3.60 show-default-prefix

`--show-default-prefix` [Command-line option]
Shows the default prefix used. This should logically be the value passed to the GNU Autoconf `./configure` script when building the game. Most other path are deduced from this one. Default is `'/usr/local'`.

4.3.61 show-default-script-file

`--show-default-script-file` [Command-line option]
Shows the default main script file path. This file is very important, since the program is more or less a hudge scheme interpreter, and this file is the file loaded by Guile. In short, it is the main program. Default is `'/usr/local/share/liquidwar6-<version>/script/liquidwar6.scm'`.

4.3.62 show-default-user-dir

`--show-default-user-dir` [Command-line option]
Shows the default user directory path. This is where run-time data, config files, log files, are stored. Default is `'$HOME/.liquidwar6/'`.

4.3.63 show-log-file

`--show-log-file` [Command-line option]
Shows the log file path. Default is `'$HOME/.liquidwar6/log.csv'`.

4.3.64 show-map-dir

`--show-map-dir` [Command-line option]
Shows the map directory. This is where builtin maps are stored. Default is `'/usr/local/share/liquidwar6-<version>/map'`.

4.3.65 show-map-path

`--show-map-path` [Command-line option]
Shows the map search path. This is where the game searches for maps. It's the combination of command-line arguments and builtin paths. Might return more directories than the one specified in a single `'map-path=dir1:dir2'` argument.

4.3.66 show-mod-dir

`--show-mod-dir` [Command-line option]
Shows the module directory path. This is where all dynamically loaded modules are stored. Default is `'/usr/local/lib/liquidwar6-<version>'`.

4.3.67 show-music-dir

`--show-music-dir` [Command-line option]
Shows the music directory. This is where builtin maps are stored. Default is `'/usr/local/share/liquidwar6-<version>/music'`.

4.3.68 show-music-path

`--show-music-path` [Command-line option]
Shows the music search path. This is where the game searches for musics. It's the combination of command-line arguments and builtin paths. Might return more directories than the one specified in a single `'music-path=dir1:dir2'` argument.

4.3.69 show-prefix

`--show-prefix` [Command-line option]
Shows the prefix used. This should logically be the value passed to the GNU Autoconf `./configure` script when building the game. Most other path are deduced from this one. Default is `'/usr/local'`.

4.3.70 show-run-dir

`--show-run-dir` [Command-line option]
Shows the run directory, usually the path where the binary is. It depends on how and where the program is launched. It is guessed from the `argc/argv` values at runtime.

4.3.71 show-script-file

`--show-script-file` [Command-line option]
Shows the main script file path. This file is very important, since the program is more or less a hudge scheme interpreter, and this file is the file loaded by Guile. In short, it is the main program. Default is `'/usr/local/share/liquidwar6-<version>/script/liquidwar6.scm'`.

4.3.72 show-user-dir

`--show-user-dir` [Command-line option]
Shows the user directory path. This is where run-time data, config files, log files, are stored. Default is `'$HOME/.liquidwar6/'`.

4.4 Path options

4.4.1 config-file

`--config-file` [Command-line option]
`LW6_CONFIG_FILE` [Environment variable]
 Type: string
 Default value: `$HOME/.liquidwar6/config.xml`
 Set the config file path. This enables you to use whatever config file you like, keeping all other informations in the same place.

4.4.2 data-dir

`--data-dir` [Command-line option]
`LW6_DATA_DIR` [Environment variable]
 Type: string
 Default value: `/usr/local/share/liquidwar6-<version>/data`
 Set the data directory. By changing this value you'll be able to use an alternative data directory.

4.4.3 log-file

`--log-file=<value>` [Command-line option]
`LW6_LOG_FILE` [Environment variable]
`log-file` [XML key]
 Type: string
 Default value: `$HOME/.liquidwar6/log.csv`
 Set the log file path. This enables you to use whatever log file you like, keeping all other informations in the same place.

4.4.4 map-dir

`--map-dir` [Command-line option]
`LW6_MAP_DIR` [Environment variable]
 Type: string
 Default value: `/usr/local/share/liquidwar6-<version>/map`
 Set the map directory path. By changing this value you'll be able to play with your own maps in your own directory. Note that there are other ways to achieve that, but using this option will work. However, a side effect is that you might not see builtin maps anymore.

4.4.5 map-path

`--map-path=<value>` [Command-line option]
`LW6_MAP_PATH` [Environment variable]
`map-path` [XML key]
 Type: string
 Default value: `$HOME/.liquidwar6/map:/usr/local/share/liquidwar6-<version>/map`

Set the map search path. By changing this value you'll be able to play with your own maps in your own directory. This is different from 'map-dir', since it includes 'map-dir', plus it adds a number of other search paths. Unlike most other parameters, the values given from the command-line, from the environment variables, or from the config file, are not overwritten, but appended. That is to say if you specify a 'map-path' with the command-line argument 'map-path=path', but also define the 'LW6_MAP_PATH' value and finally edit 'config.xml' to change the 'map-path' entry in it, you'll end up with the game searching for maps in all these directories. Additionally, 'map-dir' and '<user-dir>/map' will always be in the list. Any given value can itself include several paths, separated by the path separator. This separator is ':' on GNU/Linux, and ';' on Microsoft Windows. For instance, on a GNU/Linux box, you could use the command-line argument 'map-path=/foo/bar/map:/home/user/map:/map'.

4.4.6 mod-dir

--mod-dir [Command-line option]
 LW6_MOD_DIR [Environment variable]
 Type: string

Default value: /usr/local/lib/liquidwar6-<version>

Set the module directory path. By changing this you will load dynamic shared libraries (game specific modules such as the graphical backend) from an alternative place. Use this at your own risks, for there can always be a binary incompatibility. You've been warned.

4.4.7 music-dir

--music-dir=<value> [Command-line option]
 LW6_MUSIC_DIR [Environment variable]
 music-dir [XML key]
 Type: string

Default value: /usr/local/share/liquidwar6-<version>/music

Set the music directory path. By changing this value you'll be able to use your own musics in your own directory. Note that there are other ways to achieve that, but using this option will work. The major side effect is that using this option, you really replace the existing builtin musics by your own. If you simply want to add musics you can store them in \$HOME/.liquidwar6/music or in the map directory itself.

4.4.8 music-path

--music-path=<value> [Command-line option]
 LW6_MUSIC_PATH [Environment variable]
 music-path [XML key]
 Type: string

Default value: \$HOME/.liquidwar6/music:/usr/local/share/liquidwar6-<version>/music

Set the music search path. By changing this value you'll be able to play with your own musics in your own directory. This is different from 'music-dir', since it includes 'music-dir', plus it adds a number of other search paths. Unlike most other parameters, the values given from the command-line, from the environment variables, or from the config file, are not overwritten, but appended. That is to say if you specify a 'music-path' with the command-line argument 'music-path=path', but also define the 'LW6_MUSIC_PATH' value and finally edit 'config.xml' to change the 'music-path' entry in it, you'll end up with the game searching for musics in all these directories. Additionnally, 'music-dir' and '<user-dir>/music' will always be in the list. Any given value can itself include several pathes, separated by the path separator. This separator is ':' on GNU/Linux, and ';' on Microsoft Windows. For instance, on a GNU/Linux box, you could use the command-line argument 'music-path=/foo/bar/music:/home/user/music/./music'.

4.4.9 prefix

--prefix [Command-line option]
LW6_PREFIX [Environment variable]
 Type: string
 Default value: /usr/local

Override the prefix value given to the GNU Autoconf ./configure script when building the game. Not all path will be changed, some of them might remain the same, for instance message translations (localedir). But most game-specific data including maps, graphics, sounds, will be searched according to the new given parameter.

4.4.10 script-file

--script-file [Command-line option]
LW6_SCRIPT_FILE [Environment variable]
 Type: string
 Default value: /usr/local/share/liquidwar6-<version>/script/liquidwar6.scm

Set the main script file path. This file is very important, since the program is more or less a hudge scheme interpreter, and this file is the file loaded by Guile. In short, it is the main program.

4.4.11 user-dir

--user-dir=<value> [Command-line option]
LW6_USER_DIR [Environment variable]
user-dir [XML key]
 Type: string
 Default value: \$HOME/.liquidwar6

Set the user directory path. This is where run-time data, config files, log files, are stored. If you override this value, other parameters such as where the config and log files reside, will change.

4.5 Graphics options

4.5.1 capture

`--capture=<value>` [Command-line option]
`LW6_CAPTURE` [Environment variable]
`capture` [XML key]
Type: boolean
Default value: false
Enables capture mode, in which a BMP file is dumped on the disk (in your user directory, search for a 'capture' sub-directory).

4.5.2 fullscreen

`--fullscreen=<value>` [Command-line option]
`LW6_FULLSCREEN` [Environment variable]
`fullscreen` [XML key]
Type: boolean
Default value: false
Force the game to run fullscreen. Note that the graphics backend might ignore this hint.

4.5.3 gfx-backend

`--gfx-backend=<value>` [Command-line option]
`LW6_GFX_BACKEND` [Environment variable]
`gfx-backend` [XML key]
Type: string
Default value: gl
Sets the graphics backend AKA 'gfx' to use. For now the only choice is 'gl' and will use an OpenGL/SDL 3D-accelerated driver.

4.5.4 gfx-quality

`--gfx-quality=<value>` [Command-line option]
`LW6_GFX_QUALITY` [Environment variable]
`gfx-quality` [XML key]
Type: integer
Default value: 1 Min value: 0 Max value: 2
Sets the overall quality of the graphics backend. Depending on the backend, this can mean different things. For instance for the 'gl' backend, this can change texture filtering (nearest, linear, bilinear...). This is not the same as 'pixelize' which is a per-map option and emulates an old school appearance.

4.5.5 height

--height=<value> [Command-line option]
LW6_HEIGHT [Environment variable]
height [XML key]
 Type: integer
 Default value: -1
 Run the game with the given screen height. Note that the graphics backend might ignore this hint. Use with its companion option 'width'. A negative value will force the use of a default value.

4.5.6 width

--width=<value> [Command-line option]
LW6_WIDTH [Environment variable]
width [XML key]
 Type: integer
 Default value: -1
 Run the game with the given screen width. Note that the graphics backend might ignore this hint. Use with its companion option 'height'. A negative value will force the use of a default value.

4.5.7 windowed-mode-limit

--windowed-mode-limit=<value> [Command-line option]
LW6_WINDOWED_MODE_LIMIT [Environment variable]
windowed-mode-limit [XML key]
 Type: float
 Default value: 0.95
 When switching back from fullscreen mode to windowed mode, if we're in maximum resolution, then this coefficient will be applied before resizing the window. The idea is that (obviously) a windowed mode is preferred when a little smaller than totally fullscreen. So set this to a value just below 1.0.

4.6 Sound options

4.6.1 ambiance-exclude

--ambiance-exclude=<value> [Command-line option]
LW6_AMBIANCE_EXCLUDE [Environment variable]
ambiance-exclude [XML key]
 Type: string
 Default value:
 If this string is present in a music file name, this file won't be played during the menus, it will be excluded from the list.

4.6.2 **ambiance-file**

--ambiance-file=<value> [Command-line option]
LW6_AMBIANCE_FILE [Environment variable]
ambiance-file [XML key]
 Type: string
 Default value:
 A music file which will be used to be played during the menus. If not found, game will fallback on random files.

4.6.3 **ambiance-filter**

--ambiance-filter=<value> [Command-line option]
LW6_AMBIANCE_FILTER [Environment variable]
ambiance-filter [XML key]
 Type: string
 Default value: Chadburn
 A music filter, used to select the files which are played while navigating in the menus. It works like 'music-filter' except this one is not related to a peculiar map. This is not a complex regex-enabled filter, just a plain string search. Even the '*' wildcard won't work.

4.6.4 **music-volume**

--music-volume=<value> [Command-line option]
LW6_MUSIC_VOLUME [Environment variable]
music-volume [XML key]
 Type: float
 Default value: 0.6 Min value: 0 Max value: 1
 Set the music volume. This is a floating point value. 0 is mute. Maximum value is 1.

4.6.5 **snd-backend**

--snd-backend=<value> [Command-line option]
LW6_SND_BACKEND [Environment variable]
snd-backend [XML key]
 Type: string
 Default value: ogg
 Sets the sound backend AKA 'snd' to use. Can be 'ogg' or 'csound' but only 'ogg' will produce sound in the current release.

4.6.6 **sound-volume**

--sound-volume=<value> [Command-line option]
LW6_SOUND_VOLUME [Environment variable]
sound-volume [XML key]
 Type: float

Default value: 0.3 Min value: 0 Max value: 1

Set the sound volume. This is a floating point value. 0 is mute. Maximum value is 1.

4.7 Network options

4.7.1 bind-ip

`--bind-ip=<value>` [Command-line option]

`LW6_BIND_IP` [Environment variable]

`bind-ip` [XML key]

Type: string

Default value: 0.0.0.0

The IP address to bind on when listening to network messages. You can use this to specifically use a given network interface, the default will listen on any available interface.

4.7.2 bind-port

`--bind-port=<value>` [Command-line option]

`LW6_BIND_PORT` [Environment variable]

`bind-port` [XML key]

Type: integer

Default value: 8056 Min value: 1 Max value: 65535

The IP port to bind on when listening to network messages. The default should work out of the box, and will ease up the discovery process. That is, if you use your own settings, automatic detection of your server by other servers might not work so well.

4.7.3 cli-backends

`--cli-backends=<value>` [Command-line option]

`LW6_CLI_BACKENDS` [Environment variable]

`cli-backends` [XML key]

Type: string

Default value: tcp,udp,http

The client backends to use. Most of the time the default is fine, change it only if you specifically want to disactivate some protocol, or if you want to activate a custom-made client backend. It's a comma separated list.

4.7.4 known-nodes

`--known-nodes=<value>` [Command-line option]

`LW6_KNOWN_NODES` [Environment variable]

`known-nodes` [XML key]

Type: string

Default value: <http://ufoot.org:8056/>,<http://ufoot.hd.free.fr:8056/>

List of known nodes, nodes which the program will try to contact first to get the list of other nodes. This is mostly usefull when program is launched for the first time,

after this it should keep an up-to-date list of known servers in its internal database and automatically reconnect to them next time it starts. You might want to change this if you really want to connect to a given server which is not publically listed. The list is comma separated.

4.7.5 node-description

`--node-description=<value>` [Command-line option]
`LW6_NODE_DESCRIPTION` [Environment variable]
`node-description` [XML key]

Type: string

Default value: No description.

The description of your node, that is a text that describes your server. This will typically appear when pointing a web client on the public server URL, it is for general information, so if there's something special about your server, say it here.

4.7.6 node-title

`--node-title=<value>` [Command-line option]
`LW6_NODE_TITLE` [Environment variable]
`node-title` [XML key]

Type: string

Default value:

The title of your node, that is the name which will be displayed when listing servers. This is different from player name, for there can be several players on a single computer. By default this will be set to hostname.

4.7.7 password

`--password=<value>` [Command-line option]
`LW6_PASSWORD` [Environment variable]
`password` [XML key]

Type: string

Default value:

The password to use for network games. Do not use a valuable password, as this is stored as clear text on your hard drive. Still, the game will only send a hash/checksum of the password on the network so eavesdropper won't be able to read it. They can see the hash/checksum and use it if clever, but they can't guess the real password. A blank password means anyone can join your games when you act like a server.

4.7.8 public-url

`--public-url=<value>` [Command-line option]
`LW6_PUBLIC_URL` [Environment variable]
`public-url` [XML key]

Type: string

Default value:

The public URL of your server. By default the game will pick up one for you. In fact, the clients discovering your server should guess the public URL, probably `http://<your-ip>:<your-port>/` but you might need to use your own settings if you are using NAT or an Apache reverse-proxy to rewrite HTTP requests.

4.7.9 skip-network

`--skip-network=<value>` [Command-line option]
`LW6_SKIP_NETWORK` [Environment variable]
`skip-network` [XML key]
 Type: boolean
 Default value: true
 If set, then game won't do anything network related. No listen, no connect, no nothing. You are playing locally.

4.7.10 srv-backends

`--srv-backends=<value>` [Command-line option]
`LW6_SRV_BACKENDS` [Environment variable]
`srv-backends` [XML key]
 Type: string
 Default value: tcpd,udp,httpd
 The server backends to use. Most of the time the default is fine, change it only if you specifically want to disactivate some protocol, or if you want to activate a custom-made server backend. It's a comma separated list.

4.8 Map parameters

4.8.1 chosen-map

`--chosen-map=<value>` [Command-line option]
`LW6_CHOSEN_MAP` [Environment variable]
`chosen-map` [XML key]
 Type: string
 Default value: subflower
 The last map chosen by the player, locally. This is the map which will be used for a quick-start game, a local game, or a game started as a server.

4.8.2 force

`--force=<value>` [Command-line option]
`LW6_FORCE` [Environment variable]
`force` [XML key]
 Type: string
 Default value: respawn-team,color-conflict-mode
 A comma separated list of options which should be ignored when reading map XML files. For instance, if this contains 'rounds-per-sec,moves-per-round' then whatever

values were defined for this in 'rules.xml', then game will ignore them and use the user's values, stored in 'config.xml', running the game at the requested speed. This ultimately allows the player to control everything despite the values set by the map designer.

4.8.3 use-hints-xml

`--use-hints-xml=<value>` [Command-line option]
`LW6_USE_HINTS_XML` [Environment variable]
`use-hints-xml` [XML key]
 Type: boolean
 Default value: true
 If set, then hints will be picked up from the map defined hints.xml, if it exists. This is the default.

4.8.4 use-music-file

`--use-music-file=<value>` [Command-line option]
`LW6_USE_MUSIC_FILE` [Environment variable]
`use-music-file` [XML key]
 Type: boolean
 Default value: true
 If set, then the program will use the 'music-file' attribute to choose the music to play. If unset, then a random builtin music will be picked up, regardless of what is specified in 'music-file'.

4.8.5 use-rules-xml

`--use-rules-xml=<value>` [Command-line option]
`LW6_USE_RULES_XML` [Environment variable]
`use-rules-xml` [XML key]
 Type: boolean
 Default value: true
 If set, then rules will be picked up from the map defined rules.xml, if it exists. This is the default. Use force-time and force-size to override this and use user-defined values anyway.

4.8.6 use-style-xml

`--use-style-xml=<value>` [Command-line option]
`LW6_USE_STYLE_XML` [Environment variable]
`use-style-xml` [XML key]
 Type: boolean
 Default value: true
 If set, then style will be picked up from the map defined style.xml, if it exists. This is the default. Use force-time and force-background to override this and use user-defined values anyway.

4.8.7 use-texture

`--use-texture=<value>` [Command-line option]
`LW6_USE_TEXTURE` [Environment variable]
`use-texture` [XML key]
 Type: boolean
 Default value: true

Defines whether the map texture should be used. Of course if there's no map texture, the texture... won't be used. But if there is one, this parameter will force the game to ignore it and play with solid colors. This probably won't look as nice as the textured map in most cases, but some players might find it more readable and comfortable to play when throwing eye candy away.

4.9 Map rules.xml

4.9.1 boost-power

`--boost-power=<value>` [Command-line option]
`LW6_BOOST_POWER` [Environment variable]
`boost-power` [XML key]
 Type: integer
 Default value: 3 Min value: 0 Max value: 10

Defines how fast and powerful the boost is. That is, if on 'boost.png' it's pitch black and this parameter is set to 3, then fighters will move and act 3 times than what they would do normally.

4.9.2 color-conflict-mode

`--color-conflict-mode=<value>` [Command-line option]
`LW6_COLOR_CONFLICT_MODE` [Environment variable]
`color-conflict-mode` [XML key]
 Type: integer
 Default value: 1 Min value: 0 Max value: 2

How to handle color conflicts, that is, when a player requests a color, but this color is already used, what should be done? If 0, whether a color already exists won't affect the color of a new cursor. If 1, then two players on the same computer will be allowed to share the same color/team, but if another computer is already playing with a color, any new computer will need to use another team. If 2, then it's impossible for a new cursor to use a pre-existing color, any new cursor will require a new color, if that color is already used, a new color will be picked randomly.

4.9.3 cursor-pot-init

`--cursor-pot-init=<value>` [Command-line option]
`LW6_CURSOR_POT_INIT` [Environment variable]
`cursor-pot-init` [XML key]
 Type: integer

Default value: 100000 Min value: 5000 Max value: 500000

Defines the cursor potential at startup. Not really any reason to change it. Theoretically, there could be maps where the default value doesn't fit, but none has been seen yet.

4.9.4 danger-power

`--danger-power=<value>` [Command-line option]

`LW6_DANGER_POWER` [Environment variable]

`danger-power` [XML key]

Type: integer

Default value: 200 Min value: 0 Max value: 10000

Defines how dangerous are the black zones defined in 'danger.png'. The value is used to decrease the fighter health at each move, so you should compare its value to something like 'fighter-attack'. Being on a dangerous zone is a bit like being attacked by an invisible and unknown ennemy.

4.9.5 fighter-attack

`--fighter-attack=<value>` [Command-line option]

`LW6_FIGHTER_ATTACK` [Environment variable]

`fighter-attack` [XML key]

Type: integer

Default value: 500 Min value: 1 Max value: 10000

Defines how hard fighters will attack others, that is, in one attack, how many life-points the attacked fighter will loose. Increasing this will cause your opponents to melt faster when you attack them. With a low value, it will take ages to take on your opponents. Different styles of game. Can radically change the gameplay.

4.9.6 fighter-defense

`--fighter-defense=<value>` [Command-line option]

`LW6_FIGHTER_DEFENSE` [Environment variable]

`fighter-defense` [XML key]

Type: integer

Default value: 50 Min value: 0 Max value: 10000

Defines how fast fighters will regenerate after an attack. When this parameter is set low, an attacked fighter, which is very dark and almost dead will take a very long time to regain energy. If the parameter is set high, it can almost instantaneously regain energy.

4.9.7 fighter-new-health

`--fighter-new-health=<value>` [Command-line option]

`LW6_FIGHTER_NEW_HEALTH` [Environment variable]

`fighter-new-health` [XML key]

Type: integer

Default value: 5000 Min value: 1 Max value: 10000

Defines how healthy fighters will be when they appear on the map. This can be either at the beginning of the game or when a fighter changes team. Setting this low will allow battlefields to switch from one side to another very fast, for freshly gained fighters will be feeble and very likely to return to their original camp. To calibrate this parameter, keep in mind that the absolute maximum health a fighter can have is always 10000 (ten-thousands).

4.9.8 fighter-regenerate

`--fighter-regenerate=<value>` [Command-line option]
`LW6_FIGHTER_REGENERATE` [Environment variable]
`fighter-regenerate` [XML key]
 Type: integer

Default value: 5 Min value: 0 Max value: 10000

Defines at which speed fighters will self-regenerate, without even begin packed together. This will allow lone fighters to regenerate a bit by hiding somewhere in the map. This is typically a low value, might even be 0.

4.9.9 frags-fade-out

`--frags-fade-out=<value>` [Command-line option]
`LW6_FRAGS_FADE_OUT` [Environment variable]
`frags-fade-out` [XML key]
 Type: integer

Default value: 100 Min value: 10 Max value: 100

When a player loses (in deathmatch mode) all player points will be multiplied by this percentage, for instance if it's 90 and player had 50 points, then player will only have 45 points, then points corresponding to the new death will be added/subtracted to its total. This is to avoid players with thousands of points in advance, and keep everyone in the race. A low value will minimize the importance of game start. This is only used in modes where frags are distributed in a proportional way.

4.9.10 frags-mode

`--frags-mode=<value>` [Command-line option]
`LW6_FRAGS_MODE` [Environment variable]
`frags-mode` [XML key]
 Type: integer

Default value: 2 Min value: 0 Max value: 3

Defines how points are calculated in deathmatch mode, 0 is old school simple mode. 1 is in a mode in which 1 point is attributed to every winner, and loser loses all the corresponding points (total is always 0). 2 is proportional mode, with a total of 0 kept constant, that is, losers lose as many points as attributed to winners. 3 is a mode in which at each death, winners are attributed a number of points proportional to their fighters, and losers scores remain untouched.

4.9.11 frags-to-distribute

`--frags-to-distribute=<value>` [Command-line option]
`LW6_FRAGS_TO_DISTRIBUTE` [Environment variable]
`frags-to-distribute` [XML key]
 Type: integer
 Default value: 100 Min value: 10 Max value: 1000

Defines how many points will be distributed when in deathmatch mode. When a player loses, this amount of points will be subtracted to its total, and the same amount of points will be distributed to other live players, proportionally to how many fighters they have on the battlefield.

4.9.12 glue-power

`--glue-power=<value>` [Command-line option]
`LW6_GLUE_POWER` [Environment variable]
`glue-power` [XML key]
 Type: integer
 Default value: 20 Min value: 0 Max value: 100

Defines how sticky and powerful the glue is. That is, if on 'glue.png' it's pitch black and this parameter is set to 3, then fighters will take 3 steps to do what would normally take only one step.

4.9.13 max-cursor-pot

`--max-cursor-pot=<value>` [Command-line option]
`LW6_MAX_CURSOR_POT` [Environment variable]
`max-cursor-pot` [XML key]
 Type: integer
 Default value: 1000000 Min value: 50000 Max value: 5000000

Defines the maximum cursor potential. Not really any reason to change it. Any high value should produce the same results. Low values might reveal algorithm bugs and inconsistencies.

4.9.14 max-cursor-pot-offset

`--max-cursor-pot-offset=<value>` [Command-line option]
`LW6_MAX_CURSOR_POT_OFFSET` [Environment variable]
`max-cursor-pot-offset` [XML key]
 Type: integer
 Default value: 100 Min value: 1 Max value: 10000

Defines the maximum cursor potential offset. The idea is that in some cases, the potential of a cursor can increase in burst mode, for instance to make this cursor more important than others, so that fighters rally to it, neglecting other cursors (talking about a multi-cursor controlled team). This parameter is here to limit this burst effect and avoid bugs.

4.9.15 max-nb-cursors

--max-nb-cursors=<value> [Command-line option]
LW6_MAX_NB_CURSORS [Environment variable]
max-nb-cursors [XML key]
 Type: integer
 Default value: 26 Min value: 2 Max value: 26
 Defines the maximum number of cursors who can enter the game. Really makes sense in network games. Default value is 26, the maximum.

4.9.16 max-nb-nodes

--max-nb-nodes=<value> [Command-line option]
LW6_MAX_NB_NODES [Environment variable]
max-nb-nodes [XML key]
 Type: integer
 Default value: 12 Min value: 2 Max value: 26
 Defines the maximum number of servers who can enter the game. Really makes sense in network games. Default value is 10, and should fit in most cases. Can be raised up to 26.

4.9.17 max-nb-teams

--max-nb-teams=<value> [Command-line option]
LW6_MAX_NB_TEAMS [Environment variable]
max-nb-teams [XML key]
 Type: integer
 Default value: 10 Min value: 2 Max value: 10
 Defines the maximum number of teams who can enter the game. Really makes sense in network games. Default value is 10, the maximum.

4.9.18 max-round-delta

--max-round-delta=<value> [Command-line option]
LW6_MAX_ROUND_DELTA [Environment variable]
max-round-delta [XML key]
 Type: integer
 Default value: 1000 Min value: 1 Max value: 10000
 This is the companion value of 'round-delta'. Will put an absolute limit to the delta, which (what did you think?) is of course incremented in some cases by the core algorithm. If in doubt, don't touch.

4.9.19 max-zone-size

--max-zone-size=<value> [Command-line option]
LW6_MAX_ZONE_SIZE [Environment variable]

max-zone-size [XML key]

Type: integer

Default value: 8 Min value: 1 Max value: 64

Defines the maximum zone size, which is an internal and rather technical parameter. The idea is that to optimize things, Liquid War 6 divides the battlefield in squares, where it can, and tries to make these squares as big as possible, the idea being that everywhere in this square, fighters follow the same instructions. Just a technical optimization. The problem is that setting it too high will reveal the optimization and its tradeoffs to the player, who will see the fighter behave strangely, following invisible paths. Plus, it's ugly. Depending on your tastes (speed, look'n'feel) you'll prefer something nice or something fast. Note that anyways passed a certain value, this does not optimize anything anymore. In doubt, don't touch it.

4.9.20 medicine-power

--medicine-power=<value> [Command-line option]

LW6_MEDICINE_POWER [Environment variable]

medicine-power [XML key]

Type: integer

Default value: 100 Min value: 0 Max value: 10000

Defines how fast fighter will automatically regenerate on black zones defined in 'medicine.png'. The value is used to decrease the fighter health at each move, so you should compare its value to something like 'fighter-defense'. Being on a medicined zone is a bit like being defended by an invisible and unknown friend.

4.9.21 moves-per-round

--moves-per-round=<value> [Command-line option]

LW6_MOVES_PER_ROUND [Environment variable]

moves-per-round [XML key]

Type: integer

Default value: 2 Min value: 1 Max value: 50

Defines how many times fighters move per round. Increasing this will just make fighters move faster, but won't change anything for the rest, that is keyboard and mouse responsivity, and network traffic will stay the same. Multiplying the number of moves per round by the number of rounds per second will give the number of moves per second, which is, in fact, how fast fighters move on the screen.

4.9.22 nb-attack-tries

--nb-attack-tries=<value> [Command-line option]

LW6_NB_ATTACK_TRIES [Environment variable]

nb-attack-tries [XML key]

Type: integer

Default value: 3 Min value: 1 Max value: 7

Defines how many tries a fighter will do before giving-up attacking and choosing another behavior (defense). By tries we mean: how many directions it will try. Going

North? Going North-West? Setting this to a low value will make fighters somewhat less aggressive. This idea is that they'll prefer to switch to the next option, that is, defense/regeneration, if there's no opponent right in front of them.

4.9.23 nb-defense-tries

`--nb-defense-tries=<value>` [Command-line option]
`LW6_NB_DEFENSE_TRIES` [Environment variable]
`nb-defense-tries` [XML key]

Type: integer

Default value: 1 Min value: 1 Max value: 7

Defines how many tries a fighter will do before giving-up attacking and choosing another behavior (do nothing). By tries we mean: how many directions it will try. Going North? Going North-West? Setting this to a low value, you'll need a very compact pack of fighters for regeneration to operate, else fighters will hang around unhealthy.

4.9.24 nb-move-tries

`--nb-move-tries=<value>` [Command-line option]
`LW6_NB_MOVE_TRIES` [Environment variable]
`nb-move-tries` [XML key]

Type: integer

Default value: 5 Min value: 3 Max value: 7

Defines how many tries a fighter will do before giving-up moving and choosing another behavior (attack or defense). By tries we mean: how many directions it will try. Going North? Going North-West? Setting this to a low value, your fighters will look very stubborn and always try to move in one direction, neglecting the fact that they could dodge. This can lead to queues of fighters and other strange behaviors. On the other hand, setting it too high will cause fighter to always avoid the enemy, and groups of fighters will just pass each other without any fight. Matter of taste.

4.9.25 respawn-position-mode

`--respawn-position-mode=<value>` [Command-line option]
`LW6_RESPAWN_POSITION_MODE` [Environment variable]
`respawn-position-mode` [XML key]

Type: integer

Default value: 1 Min value: 0 Max value: 2

Defines how teams are set up on the map when respawning. 0 means teams respect the pre-defined start positions. 1 means that a random position will be picked, among the existing positions. That is, red could take green's place. 2 means total randomness, teams can appear anywhere.

4.9.26 respawn-team

`--respawn-team=<value>` [Command-line option]

LW6_RESPAWN_TEAM [Environment variable]
respawn-team [XML key]

Type: integer

Default value: 1 Min value: 0 Max value: 1

Defines what to do when a team dies. If set to 0, team disappears forever, if set to 1, team reappears automatically with fresh fighters. It's a deathmatch mode, where the winner is not the one who stays alive the longest time, since it makes no real sense in this case, but the one who has died less often than others.

4.9.27 round-delta

--round-delta=<value> [Command-line option]
LW6_ROUND_DELTA [Environment variable]
round-delta [XML key]

Type: integer

Default value: 1 Min value: 0 Max value: 100

Conditions by how much the cursor potential will be incremented each time gradient is spreaded. Sounds cryptic? It is. The idea is that at each time you move your cursor of 1 pixel, theoretically, you'll need in the worst case to move of 1 more pixel to reach any point on the map. Of course this is not true but this is the default assumption, and gradient spread will fix that. Only in Liquid War 6 this is not even the worst case, for you can control your cursor with the mouse and cross walls. Whenever you cross a wall, you might have done a great distance from the fighters' point of view, if the map is a maze. Thus this parameter, which corrects things, experience shows it does give acceptable results to increase the cursor potential by more than one at each turn. Toy around with this if you find fighters take wrong paths on some given map. If in doubt, don't touch.

4.9.28 rounds-per-sec

--rounds-per-sec=<value> [Command-line option]
LW6_ROUNDS_PER_SEC [Environment variable]
rounds-per-sec [XML key]

Type: integer

Default value: 50 Min value: 1 Max value: 200

Defines the overall speed of the game. All other settings being equal, raising this value will cause the game to behave faster. Everything will be faster, except probably the display since your computer will calculate more game positions in a given time and spend more CPU time. It will also increase network traffic. Values between 10 and 50 really make sense.

4.9.29 side-attack-factor

--side-attack-factor=<value> [Command-line option]
LW6_SIDE_ATTACK_FACTOR [Environment variable]
side-attack-factor [XML key]

Type: integer

Default value: 20 Min value: 0 Max value: 100

Defines how hard fighters will attack sideways. It's an algorithm trick, fighters attack by default the opponent right in front, but if there's no fighter there, they will still try to attack someone else, maybe sideways. But doing this their attack is not as strong. This parameter enables you to tune this. This is a percentage.

4.9.30 side-defense-factor

`--side-defense-factor=<value>` [Command-line option]
`LW6_SIDE_DEFENSE_FACTOR` [Environment variable]
`side-defense-factor` [XML key]

Type: integer

Default value: 20 Min value: 0 Max value: 100

Defines how fast fighters will regenerate, when being side by side instead of being right in front of the other. This is a percentage.

4.9.31 single-army-size

`--single-army-size=<value>` [Command-line option]
`LW6_SINGLE_ARMY_SIZE` [Environment variable]
`single-army-size` [XML key]

Type: integer

Default value: 30 Min value: 1 Max value: 95

Defines the proportion of the whole available space, which will be occupied by an army at the beginning of the game. You can either imagine playing with almost empty maps, or play very crowded with almost no space left. This is a percentage, but will be multiplied by itself to get the actual surface. That is, 50 means 50%*50%, that is, a square of 1/2 the size of a square map, so it represents 25% (1/4) of the total surface.

4.9.32 spread-thread

`--spread-thread=<value>` [Command-line option]
`LW6_SPREAD_THREAD` [Environment variable]
`spread-thread` [XML key]

Type: integer

Default value: 0 Min value: 0 Max value: 1

If set to 1, the core algorithm will fire a separate thread to spread the gradient. By default this is turned off (set to 0). Consider this as an experimental feature, the program is already rather heavily threaded, turning this on will probably not offer any significant performance gain, even on SMP systems. This might change in the future.

4.9.33 spreads-per-round

`--spreads-per-round=<value>` [Command-line option]
`LW6_SPREADS_PER_ROUND` [Environment variable]

spreads-per-round [XML key]

Type: integer

Default value: 3 Min value: 1 Max value: 12

Defines how many times the gradient is spread per round. Gradient spread is a very Liquid War 6 specific feature, just remember that the more often you do it, the more accurately fighters will move. That is, you will be sure they really take the shortest path. Usually this does not have much effect, the default value should fit in most cases, but you might want to decrease it on very simple maps where the gradient is obvious, or increase it on complex maps where you want fighters to be real smart.

4.9.34 start-blue-x

--start-blue-x=<value> [Command-line option]

LW6_START_BLUE_X [Environment variable]

start-blue-x [XML key]

Type: integer

Default value: 90 Min value: 0 Max value: 100

X start position for the blue team. This is a percentage of map width, value between 0 and 100.

4.9.35 start-blue-y

--start-blue-y=<value> [Command-line option]

LW6_START_BLUE_Y [Environment variable]

start-blue-y [XML key]

Type: integer

Default value: 10 Min value: 0 Max value: 100

Y start position for the blue team. This is a percentage of map height, value between 0 and 100.

4.9.36 start-cyan-x

--start-cyan-x=<value> [Command-line option]

LW6_START_CYAN_X [Environment variable]

start-cyan-x [XML key]

Type: integer

Default value: 35 Min value: 0 Max value: 100

X start position for the cyan team. This is a percentage of map width, value between 0 and 100.

4.9.37 start-cyan-y

--start-cyan-y=<value> [Command-line option]

LW6_START_CYAN_Y [Environment variable]

start-cyan-y [XML key]

Type: integer

Default value: 10 Min value: 0 Max value: 100

Y start position for the cyan team. This is a percentage of map height, value between 0 and 100.

4.9.38 start-green-x

--start-green-x=<value> [Command-line option]
 LW6_START_GREEN_X [Environment variable]
 start-green-x [XML key]
 Type: integer
 Default value: 90 Min value: 0 Max value: 100
 X start position for the green team. This is a percentage of map width, value between 0 and 100.

4.9.39 start-green-y

--start-green-y=<value> [Command-line option]
 LW6_START_GREEN_Y [Environment variable]
 start-green-y [XML key]
 Type: integer
 Default value: 90 Min value: 0 Max value: 100
 Y start position for the green team. This is a percentage of map height, value between 0 and 100.

4.9.40 start-lightblue-x

--start-lightblue-x=<value> [Command-line option]
 LW6_START_LIGHTBLUE_X [Environment variable]
 start-lightblue-x [XML key]
 Type: integer
 Default value: 35 Min value: 0 Max value: 100
 X start position for the lightblue team. This is a percentage of map width, value between 0 and 100.

4.9.41 start-lightblue-y

--start-lightblue-y=<value> [Command-line option]
 LW6_START_LIGHTBLUE_Y [Environment variable]
 start-lightblue-y [XML key]
 Type: integer
 Default value: 90 Min value: 0 Max value: 100
 Y start position for the lightblue team. This is a percentage of map height, value between 0 and 100.

4.9.42 start-magenta-x

--start-magenta-x=<value> [Command-line option]
 LW6_START_MAGENTA_X [Environment variable]

start-magenta-x [XML key]
 Type: integer
 Default value: 65 Min value: 0 Max value: 100
 X start position for the magenta team. This is a percentage of map width, value between 0 and 100.

4.9.43 start-magenta-y

--start-magenta-y=<value> [Command-line option]
LW6_START_MAGENTA_Y [Environment variable]
start-magenta-y [XML key]
 Type: integer
 Default value: 90 Min value: 0 Max value: 100
 Y start position for the magenta team. This is a percentage of map height, value between 0 and 100.

4.9.44 start-orange-x

--start-orange-x=<value> [Command-line option]
LW6_START_ORANGE_X [Environment variable]
start-orange-x [XML key]
 Type: integer
 Default value: 65 Min value: 0 Max value: 100
 X start position for the orange team. This is a percentage of map width, value between 0 and 100.

4.9.45 start-orange-y

--start-orange-y=<value> [Command-line option]
LW6_START_ORANGE_Y [Environment variable]
start-orange-y [XML key]
 Type: integer
 Default value: 10 Min value: 0 Max value: 100
 Y start position for the orange team. This is a percentage of map height, value between 0 and 100.

4.9.46 start-pink-x

--start-pink-x=<value> [Command-line option]
LW6_START_PINK_X [Environment variable]
start-pink-x [XML key]
 Type: integer
 Default value: 10 Min value: 0 Max value: 100
 X start position for the pink team. This is a percentage of map width, value between 0 and 100.

4.9.47 start-pink-y

`--start-pink-y=<value>` [Command-line option]
`LW6_START_PINK_Y` [Environment variable]
`start-pink-y` [XML key]
 Type: integer
 Default value: 50 Min value: 0 Max value: 100
 Y start position for the pink team. This is a percentage of map height, value between 0 and 100.

4.9.48 start-position-mode

`--start-position-mode=<value>` [Command-line option]
`LW6_START_POSITION_MODE` [Environment variable]
`start-position-mode` [XML key]
 Type: integer
 Default value: 0 Min value: 0 Max value: 2
 Defines how teams are set up on the map at game startup. 0 means teams respect the pre-defined start positions. 1 means that a random position will be picked, among the existing positions. That is, red could take green's place. 2 means total randomness, teams can appear anywhere.

4.9.49 start-purple-x

`--start-purple-x=<value>` [Command-line option]
`LW6_START_PURPLE_X` [Environment variable]
`start-purple-x` [XML key]
 Type: integer
 Default value: 90 Min value: 0 Max value: 100
 X start position for the purple team. This is a percentage of map width, value between 0 and 100.

4.9.50 start-purple-y

`--start-purple-y=<value>` [Command-line option]
`LW6_START_PURPLE_Y` [Environment variable]
`start-purple-y` [XML key]
 Type: integer
 Default value: 50 Min value: 0 Max value: 100
 Y start position for the purple team. This is a percentage of map height, value between 0 and 100.

4.9.51 start-red-x

`--start-red-x=<value>` [Command-line option]
`LW6_START_RED_X` [Environment variable]

start-red-x [XML key]
 Type: integer
 Default value: 10 Min value: 0 Max value: 100
 X start position for the red team. This is a percentage of map width, value between 0 and 100.

4.9.52 start-red-y

--start-red-y=<value> [Command-line option]
LW6_START_RED_Y [Environment variable]
start-red-y [XML key]
 Type: integer
 Default value: 10 Min value: 0 Max value: 100
 Y start position for the red team. This is a percentage of map height, value between 0 and 100.

4.9.53 start-yellow-x

--start-yellow-x=<value> [Command-line option]
LW6_START_YELLOW_X [Environment variable]
start-yellow-x [XML key]
 Type: integer
 Default value: 10 Min value: 0 Max value: 100
 X start position for the yellow team. This is a percentage of map width, value between 0 and 100.

4.9.54 start-yellow-y

--start-yellow-y=<value> [Command-line option]
LW6_START_YELLOW_Y [Environment variable]
start-yellow-y [XML key]
 Type: integer
 Default value: 90 Min value: 0 Max value: 100
 Y start position for the yellow team. This is a percentage of map height, value between 0 and 100.

4.9.55 total-armies-size

--total-armies-size=<value> [Command-line option]
LW6_TOTAL_ARMIES_SIZE [Environment variable]
total-armies-size [XML key]
 Type: integer
 Default value: 60 Min value: 1 Max value: 95
 Defines the proportion of the whole available space, which can be occupied by all the armies present together. Setting this low, whenever a new team arrives on the map, fighters might be stolen to other teams, otherwise the game would get too crowded.

This allows you to play with reasonably enough fighters with 2 players, while still allowing interesting gameplay with many players. This is a percentage, but will be multiplied by itself to get the actual surface. That is, 50 means 50%*50%, that is, a square of 1/2 the size of a square map, so it represents 25% (1/4) of the total surface.

4.9.56 total-time

`--total-time=<value>` [Command-line option]
`LW6_TOTAL_TIME` [Environment variable]
`total-time` [XML key]
 Type: integer

Default value: 900 Min value: 10 Max value: 864000

Defines the maximum time of the game, in seconds. Note that in some cases, the game can end much earlier if some player has managed to win before the bell rings. Also, technically, this value will be translated into rounds and moves, and the game engine will wait until enough rounds and moves have been played. So if the computer is too slow and the desired speed is not reached, then the game will last for a longer time.

4.9.57 vertical-move

`--vertical-move=<value>` [Command-line option]
`LW6_VERTICAL_MOVE` [Environment variable]
`vertical-move` [XML key]
 Type: integer

Default value: 1 Min value: 0 Max value: 7

Defines when to process a vertical move (along the Z 'depth' axis). If set to 0, fighters never spontaneously move along this axis. If set to 1, it will be tried just after the first move failed. If set to 2, it will be tried just after the second move failed. And so on.

4.9.58 x-polarity

`--x-polarity=<value>` [Command-line option]
`LW6_X_POLARITY` [Environment variable]
`x-polarity` [XML key]
 Type: integer

Default value: 0 Min value: -1 Max value: 1

Defines how the map will be wrapped on the X (horizontal) axis. If set to 0, nothing is wrapped. If set to 1, the right and left borders are connected, any fighter can disappear on the right border and reappear on the left border, for instance. If set to -1, it will be wrapped but also inversed, that is on a 320x240 map, a fighter disappearing on the left border at position (0,60) will reappear on the right border at position (319,180). You can combine it with 'y-polarity'.

4.9.59 y-polarity

`--y-polarity=<value>` [Command-line option]
`LW6_Y_POLARITY` [Environment variable]
`y-polarity` [XML key]
 Type: integer
 Default value: 0 Min value: -1 Max value: 1

Defines how the map will be wrapped on the Y (vertical) axis. If set to 0, nothing is wrapped. If set to 1, the top and bottom borders are connected, any fighter can disappear on the top border and reappear on the bottom border, for instance. If set to -1, it will be wrapped but also inversed, that is on a 320x240 map, a fighter disappearing on the bottom border at position (40,239) will reappear on the top border at position (280,0). You can combine it with 'x-polarity'.

4.9.60 z-polarity

`--z-polarity=<value>` [Command-line option]
`LW6_Z_POLARITY` [Environment variable]
`z-polarity` [XML key]
 Type: integer
 Default value: 0 Min value: 0 Max value: 1

Defines how the map will be wrapped on the Z (deep) axis. If set to 0, nothing is wrapped. If set to 1, when using a 4 layer map, for instance, fighters on layer 1 will be able to go directly to layer 4 even if layers 2 and 3 are filled with walls. A value of -1 is forbidden, this is not like x and y axis, it does not really make sense. Consider this an advanced setting which might save a layer in some tricky cases, the default value of 0 should fit in most cases.

4.10 Map hints.xml

4.10.1 background-color-auto

`--background-color-auto=<value>` [Command-line option]
`LW6_BACKGROUND_COLOR_AUTO` [Environment variable]
`background-color-auto` [XML key]
 Type: boolean
 Default value: true

Defines whether hud colors will be set automatically from base and alternate colors. This is a time saver to keep map designers from requiring to redefine every single color in the game. You only need to set `color-base-bg`, `color-base-fg`, `color-alternate-bg` and `color-alternate-fg`. Then `hud_color_frame_bg`, `hud_color_frame_fg`, `hud_color_text_bg` and `hud_color_text_fg` will be automatically set.

4.10.2 downsize-using-bench-value

`--downsize-using-bench-value=<value>` [Command-line option]
`LW6_DOWNSIZE_USING_BENCH_VALUE` [Environment variable]

downsize-using-bench-value [XML key]

Type: boolean

Default value: true

If set, then the game will automatically downsize a map according to the 'bench-value' parameter. Downsizing means: a 1600x1200 maps becomes 200x150, for instance. Downsizing causes fighters to be bigger because map resolution is lower. This will avoid running the game on a too big map, with your computer not being able to handle it at the required speed.

4.10.3 downsize-using-fighter-scale

--downsize-using-fighter-scale=<value> [Command-line option]

LW6_DOWNSIZE_USING_FIGHTER_SCALE [Environment variable]

downsize-using-fighter-scale [XML key]

Type: boolean

Default value: false

If set, then the game will automatically downsize a map according to the 'fighter-scale' parameter. Downsizing means: a 1600x1200 maps becomes 200x150, for instance. Downsizing causes fighters to be bigger because map resolution is lower. This can be usefull if you don't want fighters to be too small.

4.10.4 fighter-scale

--fighter-scale=<value> [Command-line option]

LW6_FIGHTER_SCALE [Environment variable]

fighter-scale [XML key]

Type: float

Default value: 1.0

Defines how wide (in pixels) fighters must be. This parameter is very important and will largely condition the number of fighters on the map. It is used when loading the map. If it is, for instance, set to 1, there will be exactly a fighter per pixel on the screen. That is, if you play 640x480 on an empty map, the maximum fighters you could have is about 300000. The idea is that by changing the resolution, you also define the density of the map. In pratice, this is done in the hope that someone with a slow computer will pick up a low resolution and therefore play small levels. Conversely, someone with a brand new computer with powerfull CPU & GPU will use great resolutions and be happy with many fighters on the map. Still, changing the resolution after loading the map will not affet the number of fighters. Same for network games, the first player, who loads the map, defines its properties according to its own settings.

4.10.5 guess-colors

--guess-colors=<value> [Command-line option]

LW6_GUESS_COLORS [Environment variable]

guess-colors [XML key]

Type: boolean

Default value: true

Defines whether colors should be set automatically from texture colors. If set to true, then the program will try to pick up colors automatically from the texture, and will override the values of the `color-base-bg`, `color-base-fg`, `color-alternate-bg` and `color-alternate-fg` parameters. How these colors are picked up can't be guaranteed, so if the map does not have strong contrast or if there can be any form of ambiguity, it's safe to set this to false and define one's own colors.

4.10.6 guess-moves-per-sec

`--guess-moves-per-sec=<value>` [Command-line option]
`LW6_GUESS_MOVES_PER_SEC` [Environment variable]
`guess-moves-per-sec` [XML key]
 Type: boolean

Default value: true

If set, then loader will use 'time-to-cross-level' to guess the game speed parameters.

4.10.7 hud-color-auto

`--hud-color-auto=<value>` [Command-line option]
`LW6_HUD_COLOR_AUTO` [Environment variable]
`hud-color-auto` [XML key]
 Type: boolean

Default value: true

Defines whether hud colors will be set automatically from base and alternate colors. This is a time saver to keep map designers from requiring to redefine every single color in the game. You only need to set `color-base-bg`, `color-base-fg`, `color-alternate-bg` and `color-alternate-fg`. Then `hud_color_frame_bg`, `hud_color_frame_fg`, `hud_color_text_bg` and `hud_color_text_fg` will be automatically set.

4.10.8 max-map-height

`--max-map-height=<value>` [Command-line option]
`LW6_MAX_MAP_HEIGHT` [Environment variable]
`max-map-height` [XML key]
 Type: integer

Default value: 3000

Allows you to give a maximum map height. When designing a map you might wonder: this is dumb I'm conceiving this map I know its height, why should I limit it? Now think of the player who plays on a old slowish computer with a tiny screen. He might redefine this himself, and does not necessarily wishes to fire Gimp to rescale the map.

4.10.9 max-map-surface

`--max-map-surface=<value>` [Command-line option]
`LW6_MAX_MAP_SURFACE` [Environment variable]

max-map-surface [XML key]

Type: integer

Default value: 3000000

Allows you to give a maximum map surface. Map surface is simply (width * height). This parameter is just here to save you the hassle of defining both 'max-map-width' and 'max-map-height' in a consistent manner.

4.10.10 max-map-width

--max-map-width=<value> [Command-line option]

LW6_MAX_MAP_WIDTH [Environment variable]

max-map-width [XML key]

Type: integer

Default value: 4000

Allows you to give a maximum map width. When designing a map you might wonder: this is dumb I'm conceiving this map I know its width, why should I limit it? Now think of the play who plays on a old slowish computer with a tiny screen. He might redefine this himself, and does not necessarily wishes to fire Gimp to rescale the map.

4.10.11 menu-color-auto

--menu-color-auto=<value> [Command-line option]

LW6_MENU_COLOR_AUTO [Environment variable]

menu-color-auto [XML key]

Type: boolean

Default value: true

Defines wether menu colors will be set automatically from base and alternate colors. This is a time saver to keep map designers from requiring to redefined every single color in the game. You only need to set color-base-bg, color-base-fg, color-alternate-bg and color-alternate-fg. Then menu_color_default_bg, menu_color_default_fg, menu_color_selected_bg, menu_color_selected_fg, menu_color_disabled_bg and menu_color_disabled_fg will be automatically set.

4.10.12 min-map-height

--min-map-height=<value> [Command-line option]

LW6_MIN_MAP_HEIGHT [Environment variable]

min-map-height [XML key]

Type: integer

Default value: 15

Allows you to give a minimum map height. When designing a map you might wonder: this is dumb I'm conceiving this map I know its height, why should I limit it? Now think of the player who decided to play with highly-defined maps because he has a super calculator and a hudge screen. He might redefine this himself, and does not necessarily wishes to fire Gimp to rescale the map.

4.10.13 min-map-surface

--min-map-surface=<value> [Command-line option]
LW6_MIN_MAP_SURFACE [Environment variable]
min-map-surface [XML key]
 Type: integer
 Default value: 1200

Allows you to give a minimum map surface. Map surface is simply (width * height). This parameter is just here to save you the hassle of defining both 'min-map-width' and 'min-map-height' in a consistent manner.

4.10.14 min-map-width

--min-map-width=<value> [Command-line option]
LW6_MIN_MAP_WIDTH [Environment variable]
min-map-width [XML key]
 Type: integer
 Default value: 20

Allows you to give a minimum map width. When designing a map you might wonder: this is dumb I'm conceiving this map I know its width, why should I limit it? Now think of the player who decided to play with highly-defined maps because he has a super calculator and a hudge screen. He might redefine this himself, and does not necessarily wishes to fire Gimp to rescale the map.

4.10.15 resample

--resample=<value> [Command-line option]
LW6_RESAMPLE [Environment variable]
resample [XML key]
 Type: boolean
 Default value: true

If set to true, maps will always be resampled to a size which depends on your screen resolution, zoom factor, and the rest. If false, maps will be set at the exact resolution of map.png.

4.10.16 speed

--speed=<value> [Command-line option]
LW6_SPEED [Environment variable]
speed [XML key]
 Type: float
 Default value: 1.0

This parameter is the main parameter on which game speed depends. The map loader will guarantee, by downscaling the map, that to cross the level (by crossing the level we mean, for instance, going from top-left corner to bottom-right corner in a straight line) a fighter will take a constant amount of time. Under the hood, the loader might

of course rescale the map but it will also change game speed so that, at the end, fighters take a constant time to cross the level. This is, indeed, the most important thing, players do not care much if internally there are X or Y moves per second, the global game experience depends on how fast fighter movement looks on the screen. The default settings corresponds roughly to one second to cross the level. If you set this to 2.0, it will go twice faster.

4.10.17 system-color-auto

`--system-color-auto=<value>` [Command-line option]
`LW6_SYSTEM_COLOR_AUTO` [Environment variable]
`system-color-auto` [XML key]
 Type: boolean
 Default value: true

Defines whether system colors will be set automatically from base and alternate colors. This is a time saver to keep map designers from requiring to redefine every single color in the game. You only need to set `color-base-bg`, `color-base-fg`, `color-alternate-bg` and `color-alternate-fg`. Then `system_color_bg` and `system_color_fg` will be automatically set.

4.10.18 upsize-using-bench-value

`--upsized-using-bench-value=<value>` [Command-line option]
`LW6_UPSIZE_USING_BENCH_VALUE` [Environment variable]
`upsized-using-bench-value` [XML key]
 Type: boolean
 Default value: false

If set, then the game will automatically upsize a map according to the 'fighter-scale' parameter. Upsizing means: a 160x120 map becomes 400x300, for instance. Upsizing causes fighters to be smaller because map resolution is higher. This will avoid useless pixelish 'jumbo fighters' look when your computer is powerful enough to do better.

4.10.19 upsize-using-fighter-scale

`--upsized-using-fighter-scale=<value>` [Command-line option]
`LW6_UPSIZE_USING_FIGHTER_SCALE` [Environment variable]
`upsized-using-fighter-scale` [XML key]
 Type: boolean
 Default value: true

If set, then the game will automatically upsize a map according to the 'fighter-scale' parameter. Upsizing means: a 160x120 map becomes 400x300, for instance. Upsizing causes fighters to be smaller because map resolution is higher. This can be useful if you don't want fighters to be too big.

4.10.20 view-color-auto

`--view-color-auto=<value>` [Command-line option]

LW6_VIEW_COLOR_AUTO [Environment variable]
view-color-auto [XML key]

Type: boolean

Default value: true

Defines whether view colors will be set automatically from base and alternate colors. This is a time saver to keep map designers from requiring to redefine every single color in the game. You only need to set `color-base-bg`, `color-base-fg`, `color-alternate-bg` and `color-alternate-fg`. Then `view_color_cursor_bg`, `view_color_cursor_fg`, `view_color_map_bg` and `view_color_map_fg` will be automatically set.

4.10.21 wall-grease

--wall-grease=<value> [Command-line option]
LW6_WALL_GREASE [Environment variable]
wall-grease [XML key]

Type: integer

Default value: 0 Min value: -5 Max value: 5

This parameter allows you to make walls (aka map foreground) thicker, or thinner, when map is loaded. Indeed, when map are resampled, and especially when they are downsampled, some walls may disappear, or some passages may be blocked. The loader can't automatically figure out whether it's more important to keep an existing wall or to keep an open passage for fighters. This parameter helps doing so, if you set it to a low value, level will be less greasy, and many passages might open themselves. On the contrary, if grease is at a high level, then a thin line of almost isolated pixels might become a thick wall. There's no real guarantee your wall or passage will always be present, but it's a same bet to assume on a 'tunnel-like' level one needs to set grease to a low value, and on a 'wide open' level with few walls one needs to set grease to a high value.

4.11 Map style.xml

4.11.1 animation-density

--animation-density=<value> [Command-line option]
LW6_ANIMATION_DENSITY [Environment variable]
animation-density [XML key]

Type: float

Default value: 1.0f Min value: 0 Max value: 10

Density of the background animation, that is, for instance, if the background animation is about displaying bubbles, using a high value will display many bubbles. A value of 1.0 corresponds to the default setting.

4.11.2 animation-speed

--animation-speed=<value> [Command-line option]
LW6_ANIMATION_SPEED [Environment variable]

animation-speed [XML key]

Type: float

Default value: 1.0f Min value: 0 Max value: 10

Speed of the background animation, that is, for instance, if the background animation is about displaying bubbles, using a high value will cause bubbles to move very fast. A value of 1.0 corresponds to the default setting.

4.11.3 background-color-root-bg

--background-color-root-bg=<value> [Command-line option]

LW6_BACKGROUND_COLOR_ROOT_BG [Environment variable]

background-color-root-bg [XML key]

Type: color

Default value: #000

Defines the main background color. This is, for instance, the color which will be used to clear the screen before drawing thing. Will be automatically guessed from the map texture if color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.4 background-color-root-fg

--background-color-root-fg=<value> [Command-line option]

LW6_BACKGROUND_COLOR_ROOT_FG [Environment variable]

background-color-root-fg [XML key]

Type: color

Default value: #ccc

Defines a color which will be used together with color-base-bg to compose the background. It can be wise to have a minimum contrast between this color and color-base-bg, but it is not mandatory, especially if other colors are manually redefined. Will be automatically guessed from the map texture if color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.5 background-color-stuff-bg

--background-color-stuff-bg=<value> [Command-line option]

LW6_BACKGROUND_COLOR_STUFF_BG [Environment variable]

background-color-stuff-bg [XML key]

Type: color

Default value: #333

Defines a color which will be used together with color-alternate-fg to draw things (animations, sprites, text, whatever) in the background. It should be different enough from color-alternate-fg so that one can really distinguish these colors. Will be automatically guessed from the map texture if color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.6 background-color-stuff-fg

`--background-color-stuff-fg=<value>` [Command-line option]
`LW6_BACKGROUND_COLOR_STUFF_FG` [Environment variable]
`background-color-stuff-fg` [XML key]
 Type: color
 Default value: #fff

Defines a color which will be used to draw things (animations, sprites, text, whatever) in the background. It should be different enough from color-alternate-bg so that one can really distinguish these colors. Think of this as the sprite, the text, the whatever-needs-to-be-seen-uses-this color. Will be automatically guessed from the map texture if color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.7 background-style

`--background-style=<value>` [Command-line option]
`LW6_BACKGROUND_STYLE` [Environment variable]
`background-style` [XML key]
 Type: string
 Default value: bubbles

The background defines, of course, what is displayed at the background, but it also conditions the colors used for other items, such as the menus for instance. The possible values are 'void' and 'bubbles'.

4.11.8 color-alternate-bg

`--color-alternate-bg=<value>` [Command-line option]
`LW6_COLOR_ALTERNATE_BG` [Environment variable]
`color-alternate-bg` [XML key]
 Type: color
 Default value: #333

Defines the alternate color, more precisely, its bg (background) part. Colors are always defined by a bg/fg pair. Most colors in the game can be deduced from this one, usually to color a map you only need to define color-base-bg, color-base-fg, color-alternate-bg and color-alternate-fg.

4.11.9 color-alternate-fg

`--color-alternate-fg=<value>` [Command-line option]
`LW6_COLOR_ALTERNATE_FG` [Environment variable]
`color-alternate-fg` [XML key]
 Type: color
 Default value: #fff

Defines the alternate color, more precisely, its fg (foreground) part. Colors are always defined by a bg/fg pair. Most colors in the game can be deduced from this one, usually to color a map you only need to define color-base-bg, color-base-fg, color-alternate-bg and color-alternate-fg.

4.11.10 color-base-bg

--color-base-bg=<value> [Command-line option]
LW6_COLOR_BASE_BG [Environment variable]
color-base-bg [XML key]
 Type: color
 Default value: #000

Defines the base color, more precisely, its bg (background) part. Colors are always defined by a bg/fg pair. Most colors in the game can be deduced from this one, usually to color a map you only need to define color-base-bg, color-base-fg, color-alternate-bg and color-alternate-fg.

4.11.11 color-base-fg

--color-base-fg=<value> [Command-line option]
LW6_COLOR_BASE_FG [Environment variable]
color-base-fg [XML key]
 Type: color
 Default value: #ccc

Defines the base color, more precisely, its fg (foreground) part. Colors are always defined by a bg/fg pair. Most colors in the game can be deduced from this one, usually to color a map you only need to define color-base-bg, color-base-fg, color-alternate-bg and color-alternate-fg.

4.11.12 colorize

--colorize=<value> [Command-line option]
LW6_COLORIZE [Environment variable]
colorize [XML key]
 Type: boolean
 Default value: true

If set, then all background drawings including textures will use the background colors. This means, for instance, that if background colors are set automatically by color-auto from the map texture, then the background will adopt the same range of colors than the map itself. In short, the background will mimic the map.

4.11.13 cursor-size

--cursor-size=<value> [Command-line option]
LW6_CURSOR_SIZE [Environment variable]
cursor-size [XML key]
 Type: float
 Default value: 1.0f Min value: 0 Max value: 10

Size of the cursors on the map. 1 is the default, setting it to a higher value will make cursors bigger, a lower value will make them smaller.

4.11.14 hidden-layer-alpha

`--hidden-layer-alpha=<value>` [Command-line option]
`LW6_HIDDEN_LAYER_ALPHA` [Environment variable]
`hidden-layer-alpha` [XML key]
 Type: float
 Default value: 0.1f Min value: 0 Max value: 1

Whenever players are supposed to be hidden behind a wall, for instance if they are in layer 2 and layer 1 is filled with walls, it's still possible to see them, but with a low alpha value (almost transparent). This parameter allows you to trick this value, 0 will make these players absolutely invisible, 1 will make them totally opaque, like if they were on layer 1.

4.11.15 hud-color-frame-bg

`--hud-color-frame-bg=<value>` [Command-line option]
`LW6_HUD_COLOR_FRAME_BG` [Environment variable]
`hud-color-frame-bg` [XML key]
 Type: color
 Default value: #000

Defines the background color for the hud frame. Ignored if hud-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.16 hud-color-frame-fg

`--hud-color-frame-fg=<value>` [Command-line option]
`LW6_HUD_COLOR_FRAME_FG` [Environment variable]
`hud-color-frame-fg` [XML key]
 Type: color
 Default value: #ccc

Defines the foreground color for the hud frame. Ignored if hud-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.17 hud-color-text-bg

`--hud-color-text-bg=<value>` [Command-line option]
`LW6_HUD_COLOR_TEXT_BG` [Environment variable]
`hud-color-text-bg` [XML key]
 Type: color
 Default value: #333

Defines the background color for hud text. Ignored if hud-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.18 hud-color-text-fg

`--hud-color-text-fg=<value>` [Command-line option]
`LW6_HUD_COLOR_TEXT_FG` [Environment variable]

hud-color-text-fg [XML key]
 Type: color
 Default value: #fff
 Defines the foreground color for hud text. Ignored if hud-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.19 hud-style

--hud-style=<value> [Command-line option]
LW6_HUD_STYLE [Environment variable]
hud-style [XML key]
 Type: string
 Default value: floating
 The hud is where informations about the game are displayed. This means, who is winning, are other status-like informations. Possible values include 'floating' and 'tactical'.

4.11.20 keep-ratio

--keep-ratio=<value> [Command-line option]
LW6_KEEP_RATIO [Environment variable]
keep-ratio [XML key]
 Type: boolean
 Default value: true
 Defines wether the map should keep its ratio, or if it should be stretched to fill the shape of your screen.

4.11.21 menu-color-default-bg

--menu-color-default-bg=<value> [Command-line option]
LW6_MENU_COLOR_DEFAULT_BG [Environment variable]
menu-color-default-bg [XML key]
 Type: color
 Default value: #333
 Defines the default background color for menus. Ignored if menu-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.22 menu-color-default-fg

--menu-color-default-fg=<value> [Command-line option]
LW6_MENU_COLOR_DEFAULT_FG [Environment variable]
menu-color-default-fg [XML key]
 Type: color
 Default value: #fff
 Defines the default foreground color for menus. In fact, this is the main color for menu text, the color used to draw letters in menus. Ignored if menu-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.23 menu-color-disabled-bg

`--menu-color-disabled-bg=<value>` [Command-line option]
`LW6_MENU_COLOR_DISABLED_BG` [Environment variable]
`menu-color-disabled-bg` [XML key]
 Type: color
 Default value: #000
 Defines the background color for a disabled menu item. Ignored if menu-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.24 menu-color-disabled-fg

`--menu-color-disabled-fg=<value>` [Command-line option]
`LW6_MENU_COLOR_DISABLED_FG` [Environment variable]
`menu-color-disabled-fg` [XML key]
 Type: color
 Default value: #ccc
 Defines the foreground color for a disabled menu item. Ignored if menu-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.25 menu-color-selected-bg

`--menu-color-selected-bg=<value>` [Command-line option]
`LW6_MENU_COLOR_SELECTED_BG` [Environment variable]
`menu-color-selected-bg` [XML key]
 Type: color
 Default value: #fff
 Defines the background color for a selected menu item. Ignored if menu-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.26 menu-color-selected-fg

`--menu-color-selected-fg=<value>` [Command-line option]
`LW6_MENU_COLOR_SELECTED_FG` [Environment variable]
`menu-color-selected-fg` [XML key]
 Type: color
 Default value: #333
 Defines the foreground color for a selected menu item. Ignored if menu-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.27 menu-style

`--menu-style=<value>` [Command-line option]
`LW6_MENU_STYLE` [Environment variable]
`menu-style` [XML key]
 Type: string
 Default value: cylinder

The menu style is simply the name of the engine used to power the menu system. The only possible value, for now, is 'cylinder'.

4.11.28 music-exclude

`--music-exclude=<value>` [Command-line option]
`LW6_MUSIC_EXCLUDE` [Environment variable]
`music-exclude` [XML key]
 Type: string
 Default value: Chadburn
 If this string is found in a music file name, it will be excluded from the list when playing in random mode.

4.11.29 music-file

`--music-file=<value>` [Command-line option]
`LW6_MUSIC_FILE` [Environment variable]
`music-file` [XML key]
 Type: string
 Default value:
 Allows you to play a custom music file (typically your own ogg music) and override default game music. If file does not exist, game will use its internal music. The file will be searched for in the current 'music-path' but also in the current map directory. No absolute or even relative path are allowed, only a plain filename with no slash or backslash. Avoid special characters at all cost.

4.11.30 music-filter

`--music-filter=<value>` [Command-line option]
`LW6_MUSIC_FILTER` [Environment variable]
`music-filter` [XML key]
 Type: string
 Default value:
 A music filter, used when files are played randomly. This is not a complex regex-enabled filter, just a plain string search. Even the '*' wildcard won't work. If you want precise control on what music file to play, please consider reorganizing your files and/or use the 'music-file' parameter.

4.11.31 pixelize

`--pixelize=<value>` [Command-line option]
`LW6_PIXELIZE` [Environment variable]
`pixelize` [XML key]
 Type: boolean
 Default value: false
 Depending on the renderer capabilities, will try to pixelize some parts of the game. This can be used to emulate the old LW5 appearance.

4.11.32 system-color-bg

--system-color-bg=<value> [Command-line option]
LW6_SYSTEM_COLOR_BG [Environment variable]
system-color-bg [XML key]
 Type: color
 Default value: #333
 Defines the system background color, used when displaying system info, such as the number of frames per second. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.33 system-color-fg

--system-color-fg=<value> [Command-line option]
LW6_SYSTEM_COLOR_FG [Environment variable]
system-color-fg [XML key]
 Type: color
 Default value: #fff
 Defines the system foreground color, used when displaying system info, such as the number of frames per second. This will typically be text color. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.34 team-color-blue

--team-color-blue=<value> [Command-line option]
LW6_TEAM_COLOR_BLUE [Environment variable]
team-color-blue [XML key]
 Type: color
 Default value: #00f
 Defines the color for the blue team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.35 team-color-cyan

--team-color-cyan=<value> [Command-line option]
LW6_TEAM_COLOR_CYAN [Environment variable]
team-color-cyan [XML key]
 Type: color
 Default value: #0ff
 Defines the color for the cyan team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.36 team-color-dead

--team-color-dead=<value> [Command-line option]
LW6_TEAM_COLOR_DEAD [Environment variable]
team-color-dead [XML key]
 Type: color
 Default value: #000

Defines the color for the teams when they are dead. By default it is black, this means when a team is weak it becomes black. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.37 team-color-green

```
--team-color-green=<value> [Command-line option]
LW6_TEAM_COLOR_GREEN [Environment variable]
team-color-green [XML key]
    Type: color
    Default value: #0f0
```

Defines the color for the green team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.38 team-color-lightblue

```
--team-color-lightblue=<value> [Command-line option]
LW6_TEAM_COLOR_LIGHTBLUE [Environment variable]
team-color-lightblue [XML key]
    Type: color
    Default value: #8bf
```

Defines the color for the light blue team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.39 team-color-magenta

```
--team-color-magenta=<value> [Command-line option]
LW6_TEAM_COLOR_MAGENTA [Environment variable]
team-color-magenta [XML key]
    Type: color
    Default value: #f0f
```

Defines the color for the magenta team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.40 team-color-orange

```
--team-color-orange=<value> [Command-line option]
LW6_TEAM_COLOR_ORANGE [Environment variable]
team-color-orange [XML key]
    Type: color
    Default value: #f80
```

Defines the color for the orange team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.41 team-color-pink

```
--team-color-pink=<value> [Command-line option]
LW6_TEAM_COLOR_PINK [Environment variable]
team-color-pink [XML key]
    Type: color
    Default value: #f8b
```

Defines the color for the pink team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.42 team-color-purple

`--team-color-purple=<value>` [Command-line option]
`LW6_TEAM_COLOR_PURPLE` [Environment variable]
`team-color-purple` [XML key]
 Type: color
 Default value: #b8f
 Defines the color for the purple team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.43 team-color-red

`--team-color-red=<value>` [Command-line option]
`LW6_TEAM_COLOR_RED` [Environment variable]
`team-color-red` [XML key]
 Type: color
 Default value: #f00
 Defines the color for the red team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.44 team-color-yellow

`--team-color-yellow=<value>` [Command-line option]
`LW6_TEAM_COLOR_YELLOW` [Environment variable]
`team-color-yellow` [XML key]
 Type: color
 Default value: #ff0
 Defines the color for the yellow team. Syntax is HTML-like, #RGB or #RRGGBB.

4.11.45 view-color-cursor-bg

`--view-color-cursor-bg=<value>` [Command-line option]
`LW6_VIEW_COLOR_CURSOR_BG` [Environment variable]
`view-color-cursor-bg` [XML key]
 Type: color
 Default value: #333
 Defines the background cursor color. Will typically be used to draw the shape of the cursor. Ignored if view-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.46 view-color-cursor-fg

`--view-color-cursor-fg=<value>` [Command-line option]
`LW6_VIEW_COLOR_CURSOR_FG` [Environment variable]
`view-color-cursor-fg` [XML key]
 Type: color
 Default value: #fff
 Defines the foreground cursor color. Will typically be used to draw text in the cursor. Ignored if view-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.47 view-color-map-bg

`--view-color-map-bg=<value>` [Command-line option]
`LW6_VIEW_COLOR_MAP_BG` [Environment variable]
`view-color-map-bg` [XML key]
 Type: color
 Default value: #000

Defines the background map color. If there's no map texture defined or if use-texture is false, this is the color of the places where armies will go. Ignored if view-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.48 view-color-map-fg

`--view-color-map-fg=<value>` [Command-line option]
`LW6_VIEW_COLOR_MAP_FG` [Environment variable]
`view-color-map-fg` [XML key]
 Type: color
 Default value: #ccc

Defines the foreground map color. If there's no map texture defined or if use-texture is false, this is the color of walls, what armies can't go through. Ignored if view-color-auto is set. Can be #RGB, #RGBA, #RRGGBB or #RRGGBBAA.

4.11.49 view-style

`--view-style=<value>` [Command-line option]
`LW6_VIEW_STYLE` [Environment variable]
`view-style` [XML key]
 Type: string
 Default value: flat

The view style conditions which renderer is used for the map, the area where fighters are displayed. This is not the graphics backend. Indeed, the graphics backend defines which technical tool one uses (which library) one runs, whether this parameter says what kind of rendering one wants.

4.11.50 x-wrap

`--x-wrap=<value>` [Command-line option]
`LW6_X_WRAP` [Environment variable]
`x-wrap` [XML key]
 Type: boolean
 Default value: true

Defines whether the map should be wrapped on the x axis. This is the companion of 'x-polarity', if no polarity is defined, map can't be wrapped, but in some cases, one might wish to have a map with polarity but without wrapping if, for instance, textures do not tile nicely.

4.11.51 y-wrap

--y-wrap=<value> [Command-line option]
LW6_Y_WRAP [Environment variable]
y-wrap [XML key]
 Type: boolean
 Default value: true

Defines whether the map should be wrapped on the y axis. This is the companion of 'y-polarity', if no polarity is defined, map can't be wrapped, but in some cases, one might wish to have a map with polarity but without wrapping if, for instance, textures do not tile nicely.

4.11.52 zoom

--zoom=<value> [Command-line option]
LW6_ZOOM [Environment variable]
zoom [XML key]
 Type: float
 Default value: 1.0f

Defines the map zoom. If lower than 1.0, map will occupy only a fraction of the screen, if greater than 1.0, some areas will be outside the screen, and the player will need to scroll through it.

4.11.53 zoom-max

--zoom-max=<value> [Command-line option]
LW6_ZOOM_MAX [Environment variable]
zoom-max [XML key]
 Type: float
 Default value: 30.0f

Defines the max map zoom. If set to a high value, you'll be able to dynamically view the map with hudge fighters, seeing only a fraction of the level.

4.11.54 zoom-min

--zoom-min=<value> [Command-line option]
LW6_ZOOM_MIN [Environment variable]
zoom-min [XML key]
 Type: float
 Default value: 0.3f

Defines the min map zoom. If set to a low value, you'll be able to dynamically view a very small, reduced map.

4.12 Advanced settings

4.12.1 bench

`--bench` [Command-line option]
`LW6_BENCH` [Environment variable]
 Runs a benchmarking test which will report an approximative performance estimation of the game on your computer.

4.12.2 bench-value

`--bench-value=<value>` [Command-line option]
`LW6_BENCH_VALUE` [Environment variable]
`bench-value` [XML key]
 Type: integer
 Default value: 20
 Contains the current bench value of the computer running the game. This is used internally to choose the right map settings. You can override this value and use your own but... use at your own risk. Pretending you have a faster computer than what you really have can lead to confusion.

4.12.3 bin-id

`--bin-id=<value>` [Command-line option]
`LW6_BIN_ID` [Environment variable]
`bin-id` [XML key]
 Type: integer
 Default value: 0
 The internal 'bin-id' value. Note that this is not necessarily equal to the value returned by 'show-build-bin-id'. When they are different, it is assumed this is because of a software upgrade.

4.12.4 bot-iq

`--bot-iq=<value>` [Command-line option]
`LW6_BOT_IQ` [Environment variable]
`bot-iq` [XML key]
 Type: integer
 Default value: 100 Min value: 0 Max value: 200
 The IQ (intelligence quotient) of bots. Typically, a value of 100 will make the bot behave normally, performing at its best. A value of 0 will just make it act the worst way it can. Values over 100 probably won't change anything compared to 100, but this truly depends on which bot backend you're running.

4.12.5 bot-speed

`--bot-speed=<value>` [Command-line option]
`LW6_BOT_SPEED` [Environment variable]
`bot-speed` [XML key]
 Type: float

Default value: 1.0

The speed of bots, 1 means normal speed, higher value will speed it up, lower will slow it down. Note that this only has an impact on bot engines, not on the game speed itself.

4.12.6 check

`--check` [Command-line option]

`LW6_CHECK` [Environment variable]

Running the game with `'-check'` is almost like running `'-test'`, the difference is that `'-check'` will not run tests which involve graphics or sound backends, so it's adapted to pure console mode. This can be usefull for automated checks on a build farm, or if you want to check things in a headless (pure console) environment.

4.12.7 commands-per-sec

`--commands-per-sec=<value>` [Command-line option]

`LW6_COMMANDS_PER_SEC` [Environment variable]

`commands-per-sec` [XML key]

Type: integer

Default value: 5

Defines the number of commands per second. When a command is generated, orders are actually sent to the game engine, for instance, `'this cursor moved there'`. So this option will affect game responsiveness, setting this to a high value will make the game more responsive but consume bandwidth on network games.

4.12.8 debug-layer-id

`--debug-layer-id=<value>` [Command-line option]

`LW6_DEBUG_LAYER_ID` [Environment variable]

`debug-layer-id` [XML key]

Type: integer

Default value: 0 Min value: 0 Max value: 6

A team id which will be used for debugging purposes, for instance when displaying gradient.

4.12.9 debug-team-id

`--debug-team-id=<value>` [Command-line option]

`LW6_DEBUG_TEAM_ID` [Environment variable]

`debug-team-id` [XML key]

Type: integer

Default value: 0 Min value: 0 Max value: 9

A team id which will be used for debugging purposes, for instance when displaying gradient.

4.12.10 demo

`--demo` [Command-line option]
`LW6_DEMO` [Environment variable]
 Start the game in demo mode. 2 bots play against each other forever.

4.12.11 dirty-read

`--dirty-read=<value>` [Command-line option]
`LW6_DIRTY_READ` [Environment variable]
`dirty-read` [XML key]
 Type: integer
 Default value: 2 Min value: 0 Max value: 2

How to handle dirty reads and locks when displaying stuff. If set to 0, there will be no dirty reads at all, a lock (mutex) will be set whenever it's needed. If set to 1, display might be done with inconsistent data, however the data itself won't be modified while displaying. If set to 2, displayed data can (and will) be modified while the rendering thread is running.

4.12.12 display-background

`--display-background=<value>` [Command-line option]
`LW6_DISPLAY_BACKGROUND` [Environment variable]
`display-background` [XML key]
 Type: boolean
 Default value: true
 Decides whether the background animation/image should be displayed at all.

4.12.13 display-console

`--display-console=<value>` [Command-line option]
`LW6_DISPLAY_CONSOLE` [Environment variable]
`display-console` [XML key]
 Type: boolean
 Default value: false
 Defines whether the interactive system console must be displayed. Note that console support must have been enabled at compilation time. It might not be available on your computer, for instance if you are running a system such as Microsoft Windows.

4.12.14 display-cursors

`--display-cursors=<value>` [Command-line option]
`LW6_DISPLAY_CURSORS` [Environment variable]
`display-cursors` [XML key]
 Type: boolean
 Default value: true
 Debugging option which can be set to 'false' to disable the display of cursors when playing.

4.12.15 display-debug-gradient

`--display-debug-gradient=<value>` [Command-line option]
`LW6_DISPLAY_DEBUG_GRADIENT` [Environment variable]
`display-debug-gradient` [XML key]
 Type: boolean
 Default value: false
 Set this to 'true' to display the gradient, this is usefull to debug the core algorithm or understand how it works.

4.12.16 display-debug-zones

`--display-debug-zones=<value>` [Command-line option]
`LW6_DISPLAY_DEBUG_ZONES` [Environment variable]
`display-debug-zones` [XML key]
 Type: boolean
 Default value: false
 Set this to 'true' to display the zones, this is usefull to debug the core algorithm or understand how it works.

4.12.17 display-fighters

`--display-fighters=<value>` [Command-line option]
`LW6_DISPLAY_FIGHTERS` [Environment variable]
`display-fighters` [XML key]
 Type: boolean
 Default value: true
 Debugging option which can be set to 'false' to disable the display of fighters when playing.

4.12.18 display-fps

`--display-fps=<value>` [Command-line option]
`LW6_DISPLAY_FPS` [Environment variable]
`display-fps` [XML key]
 Type: boolean
 Default value: false
 Set this to 'true' to display the number of frames per second. When this gets too low... play a smaller map, buy a new computer or contribute and hack Liquid War 6 so that it runs faster!

4.12.19 display-hud

`--display-hud=<value>` [Command-line option]
`LW6_DISPLAY_HUD` [Environment variable]
`display-hud` [XML key]
 Type: boolean

Default value: true

Decides whether the hud (informations while playing) should be displayed.

4.12.20 display-log

--display-log=<value> [Command-line option]

LW6_DISPLAY_LOG [Environment variable]

display-log [XML key]

Type: boolean

Default value: true

Set this to 'false' to disable the display of error messages on the screen. Note that you can miss valuable informations.

4.12.21 display-map

--display-map=<value> [Command-line option]

LW6_DISPLAY_MAP [Environment variable]

display-map [XML key]

Type: boolean

Default value: true

Debugging option which can be set to 'false' to disable map (level) display when playing.

4.12.22 display-menu

--display-menu=<value> [Command-line option]

LW6_DISPLAY_MENU [Environment variable]

display-menu [XML key]

Type: boolean

Default value: true

Debugging option which can be set to 'false' to disable the display of menus.

4.12.23 display-mouse

--display-mouse=<value> [Command-line option]

LW6_DISPLAY_MOUSE [Environment variable]

display-mouse [XML key]

Type: boolean

Default value: true

Set this to 'false' to always hide the mouse pointer.

4.12.24 display-mps

--display-mps=<value> [Command-line option]

LW6_DISPLAY_MPS [Environment variable]

display-mps [XML key]

Type: boolean

Default value: false

Set this to 'true' to display the number of moves per second. In theory the game should maintain this constant but in practise it can get low if your computer is too slow or too busy.

4.12.25 display-preview

`--display-preview=<value>` [Command-line option]

`LW6_DISPLAY_PREVIEW` [Environment variable]

`display-preview` [XML key]

Type: boolean

Default value: true

Decides wether a map preview should be displayed when choosing a level.

4.12.26 display-progress

`--display-progress=<value>` [Command-line option]

`LW6_DISPLAY_PROGRESS` [Environment variable]

`display-progress` [XML key]

Type: boolean

Default value: true

Decides wether a progress bar should be displayed when a long operation is realized as a background task.

4.12.27 display-score

`--display-score=<value>` [Command-line option]

`LW6_DISPLAY_SCORE` [Environment variable]

`display-score` [XML key]

Type: boolean

Default value: true

Decides wether the score screen should be displayed.

4.12.28 display-splash

`--display-splash=<value>` [Command-line option]

`LW6_DISPLAY_SPLASH` [Environment variable]

`display-splash` [XML key]

Type: boolean

Default value: true

Set this to 'false' to disable the display of the splash screen at game startup.

4.12.29 display-url

`--display-url=<value>` [Command-line option]

`LW6_DISPLAY_URL` [Environment variable]

display-url [XML key]

Type: boolean

Default value: false

Set this to 'true' to display the URL (homepage) of the game. This is mostly used when doing screenshots, so that generated images contain a link to the homepage.

4.12.30 executed-again

--executed-again=<value> [Command-line option]

LW6_EXECUTED_AGAIN [Environment variable]

executed-again [XML key]

Type: boolean

Default value: false

This environment variable/keyword is used to detect whether the program has been launched by itself with an internal execv call. This is used as a workaround to set some environment variables (DYLD_LIBRARY_PATH on Mac OS X for instance) before the program is run, as sometimes using setenv() inside the program does not work.

4.12.31 gfx-cpu-usage

--gfx-cpu-usage=<value> [Command-line option]

LW6_GFX_CPU_USAGE [Environment variable]

gfx-cpu-usage [XML key]

Type: float

Default value: 0.75 Min value: 0 Max value: 1

Percentage of the CPU which will be used by the display thread. It's wise to leave some time to other threads to execute. The OS does it naturally, but setting this helps the whole process by explicitly pausing (sleep call) the display thread. You could change this to a low value if you have lagging games but smooth display.

4.12.32 gfx-debug

--gfx-debug=<value> [Command-line option]

LW6_GFX_DEBUG [Environment variable]

gfx-debug [XML key]

Type: boolean

Default value: false

Enables dedicated graphics debugging tools. This is different from 'debug' mode which is global, this one is really graphics specific.

4.12.33 io-per-sec

--io-per-sec=<value> [Command-line option]

LW6_IO_PER_SEC [Environment variable]

io-per-sec [XML key]

Type: integer

Default value: 15

Defines the number of calls to input/output functions per second. This can affect speed of menus but also cursors, but won't change the speed of the game itself. It's a cosmetic, comfort option.

4.12.34 loader-sleep

`--loader-sleep=<value>` [Command-line option]

`LW6_LOADER_SLEEP` [Environment variable]

`loader-sleep` [XML key]

Type: float

Default value: 0.5

Defines how long the loader thread should wait between two polls. Default value should fit in most cases.

4.12.35 log-level

`--log-level=<value>` [Command-line option]

`LW6_LOG_LEVEL` [Environment variable]

`log-level` [XML key]

Type: integer

Default value: 3 Min value: 0 Max value: 4

Defines the log level, that is, how verbose the program will be regarding logs and console output. 0 (ERROR) is the minimum, only errors are reported. 1 (WARNING) means errors + warnings. 2 (NOTICE) displays most important messages. 3 (INFO) is the default, the log file will contain all messages but debug stuff. 4 (DEBUG) logs everything, including debug informations.

4.12.36 log-timeout

`--log-timeout=<value>` [Command-line option]

`LW6_LOG_TIMEOUT` [Environment variable]

`log-timeout` [XML key]

Type: integer

Default value: 5000

Delay, in msec, for which a log message will stay displayed on the screen.

4.12.37 magic-number

`--magic-number=<value>` [Command-line option]

`LW6_MAGIC_NUMBER` [Environment variable]

`magic-number` [XML key]

Type: integer

Default value: 33461

This 'magic' number probably requires an explanation. It's used to estimate how big a map can be built. The calculus is very approximative, basically

`bench_value*magic_number=total_fighters_on_map*rounds_per_sec*moves_per_round` with `total_fighters_on_map` depending on various parameters such as map size but also how many fighters are on the map. The map loader will try and adjust the map size so that it is just big enough not to saturate your CPU while being as high-res as possible. The magic number in itself has no real meaning, the higher it gets, the more optimized it means the game is. Normally you shouldn't change this but if you find the map resizing is too aggressively pessimistic, or if for some reason `bench` returns bogus values, you can modify it.

4.12.38 memory-bazooka-eraser

`--memory-bazooka-eraser=<value>` [Command-line option]
`LW6_MEMORY_BAZOOKA_ERASER` [Environment variable]
`memory-bazooka-eraser` [XML key]
 Type: boolean
 Default value: true

The memory eraser is a tool which will systematically fill allocated memory with 'M', and overwrite all allocated bytes with 'F' before freeing memory. It will even handle `realloc` calls. This is useful to track bugs. Indeed, with this option enabled, freshly allocated memory will never contain zeroes unless one calls `calloc`, and if you ever free some memory zone before being done with it, it will be filled with junk and therefore not be usable. The memory bazooka must be big enough if you want this feature to actually work.

4.12.39 memory-bazooka-size

`--memory-bazooka-size=<value>` [Command-line option]
`LW6_MEMORY_BAZOOKA_SIZE` [Environment variable]
`memory-bazooka-size` [XML key]
 Type: integer
 Default value: 99991

The memory bazooka is a brute-force tool, conceived after a full night spent tracking some memory leak. The idea is to keep a track of all allocated pointers, when the data was allocated (timestamp), where in the code (file, line), and even point out what data there is in that place. A memory bazooka report at the end of the game will just show what's left. There should be nothing. This parameter is here to avoid wasting CPU cycles on a feature which is very debug-oriented and does not really make sense for the casual user. Set it to 0 for best performance, something like 100 might just be helpful, but 1000000 is the right way to seriously debug code.

4.12.40 pilot-lag

`--pilot-lag=<value>` [Command-line option]
`LW6_PILOT_LAG` [Environment variable]
`pilot-lag` [XML key]
 Type: integer
 Default value: 10

Maximum lag, in rounds, until the game engine is slowed down. This will typically be usefull if your computer is too slow for the map resolution and the game speed you set up.

4.12.41 quick-start

`--quick-start` [Command-line option]
`LW6_QUICK_START` [Environment variable]
 Start the game just like if the player had requested a quick start, without showing any menu.

4.12.42 reset

`--reset` [Command-line option]
`LW6_RESET` [Environment variable]
 Clears the config file so that the game will run with defaults next time. The idea is to get rid of traces of previous executions. The difference with '-defaults' is that '-reset' does not run the game, while '-defaults' does.

4.12.43 reset-config-on-upgrade

`--reset-config-on-upgrade=<value>` [Command-line option]
`LW6_RESET_CONFIG_ON_UPGRADE` [Environment variable]
`reset-config-on-upgrade` [XML key]
 Type: boolean
 Default value: true
 If set, then a reset (config file set to defaults) is run every time you upgrade the game.

4.12.44 server

`--server` [Command-line option]
`LW6_SERVER` [Environment variable]
 Start the game in server mode, without requiring any graphics backend. Server mode is usefull if you just want to start a network node without hosting any real game on it. It can be used to list existing nodes and sessions or as a bounce server in case some clients can't contact each other because firewalled. If you only want to start a server game on your computer, don't use this option, just start the game normally and start a game server by clicking on the GUI buttons.

4.12.45 target-fps

`--target-fps=<value>` [Command-line option]
`LW6_TARGET_FPS` [Environment variable]
`target-fps` [XML key]
 Type: integer
 Default value: 60
 Defines how many frames will be displayed per second. Of course this is a maximum value, if your hardware can't keep up with this value, display will just be slow, no

matter what value you define here. Note that you might really wish to have something rather low here, to keep network and 'logic' function responsiveness. Passed 60 frames per second, speed is really only for visual comfort, as Liquid War 6 is now so fast-paced that it requires 200 frames/sec to outperform opponents.

4.13 Script hooks

4.14 C to Guile API

4.15 C functions

This section lists all documented C functions in the program. It contains many references and is self-generated from C comments using `gdoc` by [Simon Josefsson](#).

In order to reduce the number of pages of printed output, this complete reference is, by default, disabled in printable versions of the documentation (PostScript, PDF). This is both to make the manual more readable and to avoid wasting paper. Think about the environment.

It is however available in the HTML version of the documentation, which you can read online. The following addresses contain various view on the source code, giving informations on all the internal and public C interfaces:

- http://www.gnu.org/software/liquidwar6/manual/html_node/C-functions.html: the `gdoc` reference, generated from source code comments.
- <http://www.gnu.org/software/liquidwar6/coverage/>: the `lcov` output when running `./liquidwar6 --test`. It shows what functions are actually tested, and how many times they are called.
- <http://www.gnu.org/software/liquidwar6/global/>: the `GNU global` output gives complete cross-references, macros, headers, contents declaration. It's a very good place to start browsing the code.
- <http://www.gnu.org/software/liquidwar6/cyclo/>: the `pmccabe` output shows the cyclomatic complexity of functions. It enables the programmer to spots the "ugly" and dangerous parts of the program.

Appendix A 2005 .plan

Here's my .plan file, which describes what I ([Christian Mauduit](#)) have planned for Liquid War 6. There's no guarantee that what's written here is a precise description of the real future, however it should give a good idea of what I have in mind.

Note that the information here was written in summer 2005, it might or not be accurate now, as the main reason for plans to exist is that people never follow them. I'm no exception.

A.1 Complete rewrite

Liquid War 6 will be an almost complete rewrite. I mean that common code between branches 5 and 6 might end up in representing 0% of the total code. I think this is a wise decision, for the current code is really hard to maintain, and would not survive any serious cleanup. LW5 was first written in 1998, for DOS, when I had much less experience in programming. In 7 years I - and other people as well - hacked major enhancements in it such as cross-platform support, network games, and if you compare release 5.0 with the latest 5.x.x release, you'll see that a bunch of things have changed. I had never expected I would patch and fix this game for so long, and it's no surprise that it's bloated today.

FYI, here's a list of what makes LW5 unsuitable for major improvements without a complete rewrite:

- global variable hell. Lots of things are stored in globals.
- hard-coded C GUI. Read `src/level.c` to get an idea of how horrible it is.
- hard-coded 256 colors paletted mode. A clever bet in 1998 (performance...). Not anymore.
- generally bloated code. Makes bug-finding very tricky.

A.2 Technologies

Liquid War 6 will use a different technical framework than [Liquid War 5](#).

A.2.1 Script + standard C + assembly

It happens that coding a large project in pure C is a waist of time, if possible at all.

If one applies the standard 80/20 rule to a computer game, one might state that 80% of the code eat up 20% of the CPU and the other 20% of the code eat up 80% of the CPU, the former being high-level glue code and the latter being low-level algorithmic code.

With Liquid War, one could speak of the 99/01 rule. I mean that 99% of the CPU time concerns only 1% of the code, and vice-versa. Basically, Liquid War has a very CPU-greedy core algorithm, still spends a fair amount of CPU displaying stuff (but this is delegated to the low-level game programming library) and the rest is totally insignificant, in terms of CPU. Point is this "rest" represents the vast majority of the code, and also represents the very same buggy code I spend nights to patch on [Liquid War 5](#). I'm talking about network code, GUI, and other high-level glue-code which are currently being written in C.

This idea is to write all this in a convenient scripting language. There won't be any impact on performances. I can't guarantee Liquid War 6 will be blazingly fast, but for sure it won't be the scripting language fault. And of course if, as in Liquid War 3 and 5, I feel the need to implement some stuff in assembly for performances issues, I will do it.

We end up with a multi-language architecture: script + C + assembly.

My guess is that I'll use **Scheme** as an extension language. **Python** would be a good choice too. Let's say I'll give Scheme a chance, and if it's really not adapted, I'll switch back to Python. The point is that today I know Python and don't really know Scheme, but, well, it's always a pleasure for me to learn new things. It's fun.

So what is planned today is that Liquid War 6 will be a Scheme program, which will call callbacks functions written in C and/or assembly. These functions will do all the low-level time consuming algorithmic and graphical stuff. The rest of the code being entirely scripted.

A.2.2 OpenGL

Liquid War is not a 3D game, so why use OpenGL?

- it's a very convenient way to access video hardware acceleration with XFree86.
- low-end computers and/or computers without 3D acceleration can still run **Liquid War 5**.
- I'm interested in learning/using this API 8-)

This choice implies that I won't use **Allegro** anymore. Allegro stays a very convenient library and I would recommend it for it's excellent, easy to learn, powerfull, and stable. But for the needs of Liquid War 6 I'll use something else (because of OpenGL). I first thought of using **GLUT** but I might end up simply using **SDL**. The idea is just to have an OpenGL wrapper which sets up OpenGL in a similar manner on all platforms, and handles basic things such as mouse or keyboard.

A.2.3 CSound

I've got two excellent books on **Csound**, and the will to learn how to use this tool.

I'll probably use Csound for a number of things, ranging from "bubbling sounds" to full blown music. Stay tuned 8-)

A.3 Fonctionnalités

A.3.1 Visual enhancements

Of course Liquid War 6 will look nicer than **Liquid War 5**, blah blah blah. What do you think?

Maybe I'll try to use some OpenGL features to make it possible to play on a ball, on a Moebius ring, or other fancy things. I have zillion of ideas, future will decide which ones will be implemented first.

To make it clear, visual enhancements aren't my top-level priority. However I'll try and make room for these enhancements, and prepare the terrain correctly. So it's possible that the first releases of Liquid War 6 won't be that much better than **Liquid War 5**, but at least Liquid War 6 will have the possibility to evolve. Something **Liquid War 5** doesn't have.

A.3.2 Rules enhancements

There are many things that could be done easily:

- several cursors for one team

- alliances between teams
- deep places on a map, where more liquid can reside
- circular maps which "connect" the left border to the right one
- ...

As for graphical improvements, this is not my top-level priority. Simply, I'll make the game ready-to-improve. Again, all these enhancements are very hard to code in [Liquid War 5](#), else I would already have coded them. Network enhancements

That's my top-level priority.

Why is that? Well, think of Liquid War in terms of "what makes it a good game?" and "what makes it a poor game?".

It's a good game because:

- the idea is original
- the gameplay is addictive
- you can play on a LAN
- all the family can play
- it's cross-platform
- it's Free Software

It's a poor game because:

- it's somewhat ugly and has a retro "back in the eighties" look
- network games are slow on Internet
- there are not enough active Internet servers

For the ugliness, well, OpenGL and some artwork should make it. But for the network, what's the real problem?

The real problem is that in the current situation, the server needs to have all "keystrokes" before doing anything, and all players must be connected before a game starts. Here's what I plan to do to fix this:

- players will be able to connect on a game "on the fly". This is done by most online games, and it's IMHO a required features for a network mode to work on Internet (not speaking of local networks, but real wide online gaming). How this will fit with Liquid War's rules is not totally decided, but I already know of several way to achieve this.
- I'll implement an "anticipation" system "a la" [U61](#). This means that no matter if a remote player has a poor network connection, things will behave as if everything was fine. Internally, the system keeps 2 images of the game. One which is "anticipated" and displayed to the player, and one which is validated but outdated, kept internally. It's a little hard to explain, consumes twice as much CPU and memory, but it works. It happens that today the lacking ressource for playing Liquid War online is more on the network side than on the local CPU and memory aspects.
- I'll take it to the next level and implement a "peer-to-peer-like" network model, in which any client can become a server. The idea behind is that if a server quits the game, then a client takes its role, letting the game continue for hours. This way one could virtually have a never ending Liquid War game which would last weeks. I believe

this could be really cool. I also believe no proprietary game will ever implement that, for in this model there's no way to force people to access a centralized server, this server usually being the major key in the business model of a company which sells proprietary software.

This third point will be the real enhancement of Liquid War with version 6. It's one of the very points which drives me to rewrite it completely. First because it's impossible to implement it without some heavy work. Then because I find it very motivating.

A.3.3 Hey, you forgot my idea!!!

Many gamers submitted suggestions, either by mail or by posting messages on the mailing list.

Don't worry, I keep them. Not reading them here does not mean I won't implement them. It simply means I won't implement them first. I first need the game basically function before enhancing it with fancy stuff.

A.4 Road map

As I stated on the mailing list, when thinking about Liquid War 6, think of years rather than months (unless I get fired, jobless, or spend several months in a hospital with a laptop).

Note that this road map takes it for granted that I'll be the lone coder on the project. It's unlikely that someone is going to help me for the first stages, until there's at least something real, something playable. Something that proves that the concept is valid. Besides, (real) team work implies a significant overhead, especially at project start. It's hard to figure out how to distribute tasks when the tasks themselves are not clearly identified. But for the rest (starting in 2007 or 2008), it's possible that external help might greatly... ..help!

- 2005 : Project framework should be done. This implies that the scripting engine is up and running, graphical mode works, config and data loading work, basic menus are available. Nothing playable.
- 2006 : Import the core algorithm from [Liquid War 5](#), make the game playable in "demo mode" ("la" Liquid War 2), implement the network "peer-to-peer-like" mode. At this stage, it will be possible to know whether Liquid War 6 is true vaporware or not.
- 2007 : glue all this together to make something usable by anyone, heavy work on the GUI, on the options, on error checking, many bug fixes. The goal is to have a game which is equivalent to [Liquid War 5](#), with the network aspects pushed to the next level.
- 2008 : tadaaaaaaaaaaaaa! Release the game "publicly" - inform Freshmeat 8-) - and enhance it with all the feedback from gamers (bug reports and suggestions received since 1998). Work on artwork (both graphics and musics). Write documentation.
- 2009 : stabilize the game, patch it for all those things which had been forgotten back then in 2005, optimize for speed, bug-fix bug-fix bug-fix.
- 2010 : stop maintaining [Liquid War 5](#), invite Liquid War fans and coders to a hudge party in my garden, sing all night, drink beers and wine, teach Liquid War strategies to my 5 and 6 year old daughters, remember the old times when Liquid War wasn't so cool 8-)

Appendix B Fanfic

Quoting Gavin: “I wrote a liquid war fanfic some time ago [...] I wrote it after a friend claimed that there wasn’t any liquid war fanfic because it wasn’t possible.”

So here it is, a Liquid War fanfic. It was initially written for Liquid War 5, but applies to Liquid War 6 as well. Enjoy!

B.1 The Battle of Emberlificoted

...

The General presided over his massing army in his seat, or rather hovering ring, of power. It dipped slightly as he flew low over his troops marching through the viscous marsh-like terrain. They were like children: obedient, loyal, and they ate a lot.

Glancing at the status panel mounted in front of him he grimaced; the other five armies: Yellow, Green, Orange, Turquoise, and, of course, Red, were also readying armies of a similar size to his own. His violet clones would have to fight hard and eat well to win this day.

Today would not be a battle of luck, the General mused, it would be a battle of tactics, of alliances, and of betrayal. Every clone was identical - that was the general idea behind clones - and the terrain seemed strangely symmetrical; it would not give advantage to any of the six armies amassed today. Glancing at the hologram of the battlefield projected in front of him the General noted that he would have to move quickly, Orange and Yellow were too close for comfort, though fortunately Baron Red’s army of eponymous coloured clones was the furthest.

General Violet’s fingertips were sweaty even before they touched the four main control keys in front of him. They were labeled ‘W’, ‘A’, ‘D’, and, of course, the full retreat button - very useful for misleading foes and ambushing them as they pursued - ‘S’. The keys were arranged in a roughly equilateral triangular pattern; with ‘S’ forming the base and being adjacent to both ‘A’ and ‘D’, ‘W’ formed the tip of the triangle.

A long breath left his parched lips as at last he made his move.

...

“Dammit!” he screamed moments later. He had misjudged Captain Yellow and Commander Orange; he had expected one at least to attack immediately, one he could have handled. They were working together - foiling his attempt to shoot between them to near the center of the battlefield to gain a better vantage point. Yellow had shot down towards him, cutting off his advance, and now Orange had sealed his escape route. “It’s not over yet” muttered the General. He opened a voice channel with Commander Orange:

“Very clever. Flawed, but still clever.”

“Flawed?” came the reply.

“Yes flawed, when the good Captain is finished devouring my army who do you think he will turn to next?”, bluffed the General - his hands worked quickly as he manoeuvred

his hovering control ring, all that his troops ever saw of him, carefully towards the weakest section of his attackers. If he could just break out a few units he could soon turn the tide against both Yellow and Orange.

“We have an alliance...” Orange’s voice was unsure now.

Time for some sarcasm to through her even more off balance, thought the General,

“I gathered”, he spoke softly, slowly, and with too much meaning. Then closing the channel he turned his attention back to his escape.

...

“Yes!” whooped the ecstatic figure of the General. Fifty or so of his troops had broken free undetected and were even now working their way cautiously towards the camps of the Yellow army, only the front lines were still actively fighting; this opening gambit of Yellow and Orange had turned into a stale siege and Yellow’s army had pitched tent.

General Violet steered his hovering guidance ring to the center of the Yellow camp. His troops struck, both those who had got behind the lines and those who were still besieged. Yellow reacted too slowly and suddenly found that her army, was shrinking back from the onslaught. There was nowhere to run to, and bye now her only ally - Commander Orange - had abandoned her to her fate; he was too busy engaging Sir. Turquoise, who had managed to escape from the slaughter that the Baron had caused to the Turquoise ranks and was even now valiantly attacking the flanks of the Orange troops.

A glance at the status panel showed that Yellow’s life force was fading quickly: 8%, 3%, 1%, Gone.

The General smiled, he always enjoyed getting the first kill, and by now his armies life force had grown and his clones had replicated. With his, now, formidable fighting force it was no problem to engulf both Sir. Turquoise and Commander Orange’s brawling armies and annihilate them. Once again his army grew in size and power. Now if only the Baron didn’t notice that..., thought the General.

...

“Too late!” yelled the General, now thrown into panic, as he saw the approaching Baron. His army had also grown in size and power - having fatally injured the Turquoise army within the opening moments of the battle, and having finally managed to catch the elusive fleeing form of, or what remained of, Emperor Green.

Gripping the controls harder the General thought quickly, his army doesn’t so completely outnumber me that this is already over, however unless I can cause him to make a mistake that allows me to take the upper hand then I will inevitably lose. Maybe I can...

This thought was terminated and replaced by another as the Baron’s angry red troops broke through the undergrowth that had covered their movements and started to surround the General’s army. The thought that now throbbed through the panic-stricken mind of General Violet was simply ‘Run!’.

Even as he signaled the retreat and made for what seemed to be the only possible means of escape the Baron’s blood red control ring appeared at the opening. The General knew it was over, even before the host of red beings appeared at the opening.

There was no escape. His life force was almost depleted and he was surrounded. Then it was that the Baron decided to communicate:

“Too bad. It was a good game”

The General blinked, gaped, and was generally gobsmacked. Just before his life force completely failed and his own weary eyes closed in defeat he snarled,

“What!? This is not a game!” were the General’s dying words.

Appendix C Links

This section lists various Internet Liquid War related links.

- <http://www.gnu.org/software/liquidwar6/>: Liquid War 6 homepage
- <http://www.gnu.org/software/liquidwar6/manual/>: Online manual
- <http://download.savannah.gnu.org/releases/liquidwar6/>: Savannah downloads
- <http://www.ufoot.org/download/liquidwar/v6/>: ufoot.org downloads
- <http://arch.sv.gnu.org/archives/liquidwar6/>: GNU Arch repository (deprecated)
- <http://git.savannah.gnu.org/gitweb/?p=liquidwar6.git>, <http://git.savannah.gnu.org/cgit/liquidwar6.git>: GIT repository (replaces GNU Arch)
- <http://savannah.gnu.org/projects/liquidwar6/>: Project on Savannah
- <http://lists.gnu.org/archive/html/help-liquidwar6/>: Mailing-list archives
- <http://kasperhviid.dk/liquid-stuff.html>, <http://kasperhviid.dk/lw6-eyecandy.html>, <http://kasperhviid.dk/lw6-gui.html>: Kasper Hviid brainstorming and resources concerning Liquid War 6. http://en.wikipedia.org/wiki/Liquid_War: Liquid War entry on Wikipedia
- http://fr.wikipedia.org/wiki/Liquid_War: Liquid War entry on Wikipedia (French)
- http://www.lgdb.org/game/liquid_war: Liquid War 6 entry on LGDB
- http://libregamewiki.org/Liquid_War_6: Liquid War 6 entry on Libregamewiki
- <http://www.ufoot.org/liquidwar/v5/>: Liquid War 5

Appendix D GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users’ Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a. The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source.

The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

- d. Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance.

However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so

available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

program Copyright (C) year name of author
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Appendix E GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any,

- be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
 - C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
 - D. Preserve all the copyright notices of the Document.
 - E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
 - F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
 - G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their

titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts. A copy of the license is included in the section entitled ‘‘GNU
Free Documentation License’’.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix F Indexes

F.1 Concept index

A

Apple..... 31
Arch..... 40

B

Bug report..... 5, 14

C

Compilation..... 10, 27
CVS..... 42

D

Download..... 5

G

GIT..... 42
GPL, GNU General Public License..... 123

H

hints.xml..... 19

L

Log file..... 14

M

Mac OS X..... 31
Microsoft Windows..... 27

R

rules.xml..... 18

S

source code..... 42
style.xml..... 19
subversion..... 42
SVN..... 42

F.2 Function and keyword index

-	
--about=<value>.....	45
--ambiance-exclude=<value>.....	62
--ambiance-file=<value>.....	63
--ambiance-filter=<value>.....	63
--animation-density=<value>.....	89
--animation-speed=<value>.....	89
--audit.....	45
--background-color-auto=<value>.....	83
--background-color-root-bg=<value>.....	90
--background-color-root-fg=<value>.....	90
--background-color-stuff-bg=<value>.....	90
--background-color-stuff-fg=<value>.....	91
--background-style=<value>.....	91
--bench.....	102
--bench-value=<value>.....	102
--bin-id=<value>.....	102
--bind-ip=<value>.....	64
--bind-port=<value>.....	64
--boost-power=<value>.....	68
--bot-iq=<value>.....	102
--bot-speed=<value>.....	102
--capture=<value>.....	61
--check.....	103
--chosen-map=<value>.....	66
--cli-backends=<value>.....	64
--color-alternate-bg=<value>.....	91
--color-alternate-fg=<value>.....	91
--color-base-bg=<value>.....	92
--color-base-fg=<value>.....	92
--color-conflict-mode=<value>.....	68
--colorize=<value>.....	92
--commands-per-sec=<value>.....	103
--config-file.....	58
--copyright.....	45
--cursor-pot-init=<value>.....	68
--cursor-size=<value>.....	92
--danger-power=<value>.....	69
--data-dir.....	58
--debug.....	45
--debug-layer-id=<value>.....	103
--debug-team-id=<value>.....	103
--defaults.....	45
--demo.....	104
--dirty-read=<value>.....	104
--display-background=<value>.....	104
--display-console=<value>.....	104
--display-cursors=<value>.....	104

--display-debug-gradient=<value>.....	105	--list-map	47
--display-debug-zones=<value>.....	105	--list-map-hints	48
--display-fighters=<value>.....	105	--list-map-rules	48
--display-fps=<value>	105	--list-map-style	48
--display-hud=<value>	105	--list-network	48
--display-log=<value>	106	--list-path	48
--display-map=<value>	106	--list-players	48
--display-menu=<value>	106	--list-quick	48
--display-mouse=<value>	106	--list-show	48
--display-mps=<value>	106	--list-sound	48
--display-preview=<value>	107	--list-tuning	49
--display-progress=<value>	107	--loader-sleep=<value>	109
--display-score=<value>	107	--log-file=<value>.....	58
--display-splash=<value>	107	--log-level=<value>.....	109
--display-url=<value>	107	--log-timeout=<value>	109
--downsize-using-bench-value=<value>	83	--magic-number=<value>	109
--downsize-using-fighter-scale=<value>	84	--map-dir	58
--example-hints-xml	46	--map-path=<value>	58
--example-rules-xml	47	--max-cursor-pot-offset=<value>	71
--example-style-xml	47	--max-cursor-pot=<value>	71
--executed-again=<value>	108	--max-map-height=<value>	85
--fighter-attack=<value>	69	--max-map-surface=<value>	85
--fighter-defense=<value>	69	--max-map-width=<value>	86
--fighter-new-health=<value>	69	--max-nb-cursors=<value>	72
--fighter-regenerate=<value>	70	--max-nb-nodes=<value>	72
--fighter-scale=<value>	84	--max-nb-teams=<value>	72
--force=<value>	66	--max-round-delta=<value>	72
--frags-fade-out=<value>	70	--max-zone-size=<value>	72
--frags-mode=<value>	70	--medicine-power=<value>	73
--frags-to-distribute=<value>	71	--memory-bazooka-eraser=<value>	110
--fullscreen=<value>.....	61	--memory-bazooka-size=<value>	110
--gfx-backend=<value>	61	--menu-color-auto=<value>	86
--gfx-cpu-usage=<value>	108	--menu-color-default-bg=<value>	94
--gfx-debug=<value>.....	108	--menu-color-default-fg=<value>	94
--gfx-quality=<value>	61	--menu-color-disabled-bg=<value>	95
--glue-power=<value>.....	71	--menu-color-disabled-fg=<value>	95
--guess-colors=<value>	84	--menu-color-selected-bg=<value>	95
--guess-moves-per-sec=<value>	85	--menu-color-selected-fg=<value>	95
--height=<value>.....	62	--menu-style=<value>.....	95
--help	46	--min-map-height=<value>	86
--hidden-layer-alpha=<value>	93	--min-map-surface=<value>	87
--host	46	--min-map-width=<value>	87
--hud-color-auto=<value>	85	--mod-dir	59
--hud-color-frame-bg=<value>	93	--modules	46
--hud-color-frame-fg=<value>	93	--moves-per-round=<value>	73
--hud-color-text-bg=<value>	93	--music-dir=<value>.....	59
--hud-color-text-fg=<value>	93	--music-exclude=<value>	96
--hud-style=<value>.....	94	--music-file=<value>.....	96
--io-per-sec=<value>.....	108	--music-filter=<value>	96
--keep-ratio=<value>.....	94	--music-path=<value>.....	59
--known-nodes=<value>	64	--music-volume=<value>	63
--list	46	--nb-attack-tries=<value>	73
--list-aliases	47	--nb-defense-tries=<value>	74
--list-doc	47	--nb-move-tries=<value>	74
--list-funcs	47	--node-description=<value>	65
--list-graphics	47	--node-title=<value>.....	65
--list-hooks	47	--password=<value>.....	65
--list-input	47	--pedigree	46

--pilot-lag=<value>.....	110	--show-build-top-srcdir	54
--pixelize=<value>.....	96	--show-build-url.....	54
--prefix.....	60	--show-build-version.....	54
--public-url=<value>.....	65	--show-build-x86.....	54
--quick-start	111	--show-config-file.....	54
--resample=<value>.....	87	--show-cwd	55
--reset.....	111	--show-data-dir	55
--reset-config-on-upgrade=<value>	111	--show-default-config-file	55
--respawn-position-mode=<value>.....	74	--show-default-data-dir	55
--respawn-team=<value>	74	--show-default-log-file	55
--round-delta=<value>	75	--show-default-map-dir	55
--rounds-per-sec=<value>.....	75	--show-default-map-path	55
--script-file	60	--show-default-mod-dir	55
--server	111	--show-default-music-dir	56
--show-build-bin-id.....	49	--show-default-music-path	56
--show-build-cflags.....	49	--show-default-prefix	56
--show-build-codename	49	--show-default-script-file	56
--show-build-configure-args	49	--show-default-user-dir	56
--show-build-copyright	49	--show-log-file	56
--show-build-datadir	49	--show-map-dir	56
--show-build-date.....	49	--show-map-path.....	56
--show-build-docdir.....	50	--show-mod-dir	57
--show-build-enable-allnone	50	--show-music-dir	57
--show-build-enable-console	50	--show-music-path.....	57
--show-build-enable-fullstatic	50	--show-prefix	57
--show-build-enable-gcov	50	--show-run-dir	57
--show-build-enable-gprof	50	--show-script-file	57
--show-build-enable-gtk	50	--show-user-dir	57
--show-build-enable-instrument	50	--side-attack-factor=<value>	75
--show-build-enable-mod-csound	51	--side-defense-factor=<value>	76
--show-build-enable-mod-gl	51	--single-army-size=<value>	76
--show-build-enable-mod-http	51	--skip-network=<value>	66
--show-build-enable-mod-ogg	51	--snd-backend=<value>	63
--show-build-enable-optimize	51	--sound-volume=<value>	63
--show-build-enable-paranoid	51	--speed=<value>	87
--show-build-enable-profiler	51	--spread-thread=<value>	76
--show-build-enable-valgrind	52	--spreads-per-round=<value>	76
--show-build-endianness	52	--srv-backends=<value>	66
--show-build-gcc-version	52	--start-blue-x=<value>	77
--show-build-gp2x	52	--start-blue-y=<value>	77
--show-build-host-cpu	52	--start-cyan-x=<value>	77
--show-build-host-os	52	--start-cyan-y=<value>	77
--show-build-hostname	52	--start-green-x=<value>	78
--show-build-includedir	52	--start-green-y=<value>	78
--show-build-ldflags	52	--start-lightblue-x=<value>	78
--show-build-libdir.....	53	--start-lightblue-y=<value>	78
--show-build-license.....	53	--start-magenta-x=<value>	78
--show-build-localedir	53	--start-magenta-y=<value>	79
--show-build-mac-os-x	53	--start-orange-x=<value>	79
--show-build-md5sum.....	53	--start-orange-y=<value>	79
--show-build-ms-windows	53	--start-pink-x=<value>	79
--show-build-package-name	53	--start-pink-y=<value>	80
--show-build-package-string	53	--start-position-mode=<value>	80
--show-build-package-tarname	53	--start-purple-x=<value>	80
--show-build-pointer-size	54	--start-purple-y=<value>	80
--show-build-prefix.....	54	--start-red-x=<value>	80
--show-build-stamp.....	54	--start-red-y=<value>	81
--show-build-time.....	54	--start-yellow-x=<value>	81

--start-yellow-y=<value>	81
--system-color-auto=<value>	88
--system-color-bg=<value>	97
--system-color-fg=<value>	97
--target-fps=<value>	111
--team-color-blue=<value>	97
--team-color-cyan=<value>	97
--team-color-dead=<value>	97
--team-color-green=<value>	98
--team-color-lightblue=<value>	98
--team-color-magenta=<value>	98
--team-color-orange=<value>	98
--team-color-pink=<value>	98
--team-color-purple=<value>	99
--team-color-red=<value>	99
--team-color-yellow=<value>	99
--test	46
--total-armies-size=<value>	81
--total-time=<value>	82
--upsample-using-bench-value=<value>	88
--upsample-using-fighter-scale=<value>	88
--use-hints-xml=<value>	67
--use-music-file=<value>	67
--use-rules-xml=<value>	67
--use-style-xml=<value>	67
--use-texture=<value>	68
--user-dir=<value>	60
--version	46
--vertical-move=<value>	82
--view-color-auto=<value>	88
--view-color-cursor-bg=<value>	99
--view-color-cursor-fg=<value>	99
--view-color-map-bg=<value>	100
--view-color-map-fg=<value>	100
--view-style=<value>	100
--wall-grease=<value>	89
--width=<value>	62
--windowed-mode-limit=<value>	62
--x-polarity=<value>	82
--x-wrap=<value>	100
--y-polarity=<value>	83
--y-wrap=<value>	101
--z-polarity=<value>	83
--zoom-max=<value>	101
--zoom-min=<value>	101
--zoom=<value>	101

A

ambience-exclude	62
ambience-file	63
ambience-filter	63
animation-density	89
animation-speed	89

B

background-color-auto	83
-----------------------------	----

background-color-root-bg	90
background-color-root-fg	90
background-color-stuff-bg	90
background-color-stuff-fg	91
background-style	91
bench-value	102
bin-id	102
bind-ip	64
bind-port	64
boost-power	68
bot-iq	102
bot-speed	102

C

capture	61
chosen-map	66
cli-backends	64
color-alternate-bg	91
color-alternate-fg	91
color-base-bg	92
color-base-fg	92
color-conflict-mode	68
colorize	92
commands-per-sec	103
cursor-pot-init	68
cursor-size	92

D

danger-power	69
debug-layer-id	103
debug-team-id	103
dirty-read	104
display-background	104
display-console	104
display-cursors	104
display-debug-gradient	105
display-debug-zones	105
display-fighters	105
display-fps	105
display-hud	105
display-log	106
display-map	106
display-menu	106
display-mouse	106
display-mps	106
display-preview	107
display-progress	107
display-score	107
display-splash	107
display-url	107
downsize-using-bench-value	83
downsize-using-fighter-scale	84

E

executed-again	108
----------------------	-----

F

fighter-attack	69
fighter-defense	69
fighter-new-health	69
fighter-regenerate	70
fighter-scale	84
force	66
frags-fade-out	70
frags-mode	70
frags-to-distribute	71
fullscreen	61

G

gfx-backend	61
gfx-cpu-usage	108
gfx-debug	108
gfx-quality	61
glue-power	71
guess-colors	84
guess-moves-per-sec	85

H

height	62
hidden-layer-alpha	93
hud-color-auto	85
hud-color-frame-bg	93
hud-color-frame-fg	93
hud-color-text-bg	93
hud-color-text-fg	93
hud-style	94

I

io-per-sec	108
------------	-----

K

keep-ratio	94
known-nodes	64

L

loader-sleep	109
log-file	58
log-level	109
log-timeout	109
LW6_AMBIANCE_EXCLUDE	62
LW6_AMBIANCE_FILE	63
LW6_AMBIANCE_FILTER	63
LW6_ANIMATION_DENSITY	89
LW6_ANIMATION_SPEED	89
LW6_BACKGROUND_COLOR_AUTO	83
LW6_BACKGROUND_COLOR_ROOT_BG	90
LW6_BACKGROUND_COLOR_ROOT_FG	90
LW6_BACKGROUND_COLOR_STUFF_BG	90

LW6_BACKGROUND_COLOR_STUFF_FG	91
LW6_BACKGROUND_STYLE	91
LW6_BENCH	102
LW6_BENCH_VALUE	102
LW6_BIN_ID	102
LW6_BIND_IP	64
LW6_BIND_PORT	64
LW6_BOOST_POWER	68
LW6_BOT_IQ	102
LW6_BOT_SPEED	102
LW6_CAPTURE	61
LW6_CHECK	103
LW6_CHOSEN_MAP	66
LW6_CLI_BACKENDS	64
LW6_COLOR_ALTERNATE_BG	91
LW6_COLOR_ALTERNATE_FG	91
LW6_COLOR_BASE_BG	92
LW6_COLOR_BASE_FG	92
LW6_COLOR_CONFLICT_MODE	68
LW6_COLORIZE	92
LW6_COMMANDS_PER_SEC	103
LW6_CONFIG_FILE	58
LW6_CURSOR_POT_INIT	68
LW6_CURSOR_SIZE	92
LW6_DANGER_POWER	69
LW6_DATA_DIR	58
LW6_DEBUG_LAYER_ID	103
LW6_DEBUG_TEAM_ID	103
LW6_DEMO	104
LW6_DIRTY_READ	104
LW6_DISPLAY_BACKGROUND	104
LW6_DISPLAY_CONSOLE	104
LW6_DISPLAY_CURSORS	104
LW6_DISPLAY_DEBUG_GRADIENT	105
LW6_DISPLAY_DEBUG_ZONES	105
LW6_DISPLAY_FIGHTERS	105
LW6_DISPLAY_FPS	105
LW6_DISPLAY_HUD	105
LW6_DISPLAY_LOG	106
LW6_DISPLAY_MAP	106
LW6_DISPLAY_MENU	106
LW6_DISPLAY_MOUSE	106
LW6_DISPLAY_MPS	106
LW6_DISPLAY_PREVIEW	107
LW6_DISPLAY_PROGRESS	107
LW6_DISPLAY_SCORE	107
LW6_DISPLAY_SPLASH	107
LW6_DISPLAY_URL	107
LW6_DOWNSIZE_USING_BENCH_VALUE	83
LW6_DOWNSIZE_USING_FIGHTER_SCALE	84
LW6_EXECUTED_AGAIN	108
LW6_FIGHTER_ATTACK	69
LW6_FIGHTER_DEFENSE	69
LW6_FIGHTER_NEW_HEALTH	69
LW6_FIGHTER_REGENERATE	70
LW6_FIGHTER_SCALE	84
LW6_FORCE	66
LW6_FRAGS_FADE_OUT	70

LW6_FRAGS_MODE	70	LW6_MUSIC_PATH	59
LW6_FRAGS_TO_DISTRIBUTE	71	LW6_MUSIC_VOLUME	63
LW6_FULLSCREEN	61	LW6_NB_ATTACK_TRIES	73
LW6_GFX_BACKEND	61	LW6_NB_DEFENSE_TRIES	74
LW6_GFX_CPU_USAGE	108	LW6_NB_MOVE_TRIES	74
LW6_GFX_DEBUG	108	LW6_NODE_DESCRIPTION	65
LW6_GFX_QUALITY	61	LW6_NODE_TITLE	65
LW6_GLUE_POWER	71	LW6_PASSWORD	65
LW6_GUESS_COLORS	84	LW6_PILOT_LAG	110
LW6_GUESS_MOVES_PER_SEC	85	LW6_PIXELIZE	96
LW6_HEIGHT	62	LW6_PREFIX	60
LW6_HIDDEN_LAYER_ALPHA	93	LW6_PUBLIC_URL	65
LW6_HUD_COLOR_AUTO	85	LW6_QUICK_START	111
LW6_HUD_COLOR_FRAME_BG	93	LW6_RESAMPLE	87
LW6_HUD_COLOR_FRAME_FG	93	LW6_RESET	111
LW6_HUD_COLOR_TEXT_BG	93	LW6_RESET_CONFIG_ON_UPGRADE	111
LW6_HUD_COLOR_TEXT_FG	93	LW6_RESPAWN_POSITION_MODE	74
LW6_HUD_STYLE	94	LW6_RESPAWN_TEAM	74
LW6_IO_PER_SEC	108	LW6_ROUND_DELTA	75
LW6_KEEP_RATIO	94	LW6_ROUNDS_PER_SEC	75
LW6_KNOWN_NODES	64	LW6_SCRIPT_FILE	60
LW6_LOADER_SLEEP	109	LW6_SERVER	111
LW6_LOG_FILE	58	LW6_SIDE_ATTACK_FACTOR	75
LW6_LOG_LEVEL	109	LW6_SIDE_DEFENSE_FACTOR	76
LW6_LOG_TIMEOUT	109	LW6_SINGLE_ARMY_SIZE	76
LW6_MAGIC_NUMBER	109	LW6_SKIP_NETWORK	66
LW6_MAP_DIR	58	LW6_SND_BACKEND	63
LW6_MAP_PATH	58	LW6_SOUND_VOLUME	63
LW6_MAX_CURSOR_POT	71	LW6_SPEED	87
LW6_MAX_CURSOR_POT_OFFSET	71	LW6_SPREAD_THREAD	76
LW6_MAX_MAP_HEIGHT	85	LW6_SPREADS_PER_ROUND	76
LW6_MAX_MAP_SURFACE	85	LW6_SRV_BACKENDS	66
LW6_MAX_MAP_WIDTH	86	LW6_START_BLUE_X	77
LW6_MAX_NB_CURSORS	72	LW6_START_BLUE_Y	77
LW6_MAX_NB_NODES	72	LW6_START_CYAN_X	77
LW6_MAX_NB_TEAMS	72	LW6_START_CYAN_Y	77
LW6_MAX_ROUND_DELTA	72	LW6_START_GREEN_X	78
LW6_MAX_ZONE_SIZE	72	LW6_START_GREEN_Y	78
LW6_MEDICINE_POWER	73	LW6_START_LIGHTBLUE_X	78
LW6_MEMORY_BAZOOKA_ERASER	110	LW6_START_LIGHTBLUE_Y	78
LW6_MEMORY_BAZOOKA_SIZE	110	LW6_START_MAGENTA_X	78
LW6_MENU_COLOR_AUTO	86	LW6_START_MAGENTA_Y	79
LW6_MENU_COLOR_DEFAULT_BG	94	LW6_START_ORANGE_X	79
LW6_MENU_COLOR_DEFAULT_FG	94	LW6_START_ORANGE_Y	79
LW6_MENU_COLOR_DISABLED_BG	95	LW6_START_PINK_X	79
LW6_MENU_COLOR_DISABLED_FG	95	LW6_START_PINK_Y	80
LW6_MENU_COLOR_SELECTED_BG	95	LW6_START_POSITION_MODE	80
LW6_MENU_COLOR_SELECTED_FG	95	LW6_START_PURPLE_X	80
LW6_MENU_STYLE	95	LW6_START_PURPLE_Y	80
LW6_MIN_MAP_HEIGHT	86	LW6_START_RED_X	80
LW6_MIN_MAP_SURFACE	87	LW6_START_RED_Y	81
LW6_MIN_MAP_WIDTH	87	LW6_START_YELLOW_X	81
LW6_MOD_DIR	59	LW6_START_YELLOW_Y	81
LW6_MOVES_PER_ROUND	73	LW6_SYSTEM_COLOR_AUTO	88
LW6_MUSIC_DIR	59	LW6_SYSTEM_COLOR_BG	97
LW6_MUSIC_EXCLUDE	96	LW6_SYSTEM_COLOR_FG	97
LW6_MUSIC_FILE	96	LW6_TARGET_FPS	111
LW6_MUSIC_FILTER	96	LW6_TEAM_COLOR_BLUE	97

LW6_TEAM_COLOR_CYAN	97
LW6_TEAM_COLOR_DEAD	97
LW6_TEAM_COLOR_GREEN	98
LW6_TEAM_COLOR_LIGHTBLUE	98
LW6_TEAM_COLOR_MAGENTA	98
LW6_TEAM_COLOR_ORANGE	98
LW6_TEAM_COLOR_PINK	98
LW6_TEAM_COLOR_PURPLE	99
LW6_TEAM_COLOR_RED	99
LW6_TEAM_COLOR_YELLOW	99
LW6_TOTAL_ARMIES_SIZE	81
LW6_TOTAL_TIME	82
LW6_UPSIZE_USING_BENCH_VALUE	88
LW6_UPSIZE_USING_FIGHTER_SCALE	88
LW6_USE_HINTS_XML	67
LW6_USE_MUSIC_FILE	67
LW6_USE_RULES_XML	67
LW6_USE_STYLE_XML	67
LW6_USE_TEXTURE	68
LW6_USER_DIR	60
LW6_VERTICAL_MOVE	82
LW6_VIEW_COLOR_AUTO	88
LW6_VIEW_COLOR_CURSOR_BG	99
LW6_VIEW_COLOR_CURSOR_FG	99
LW6_VIEW_COLOR_MAP_BG	100
LW6_VIEW_COLOR_MAP_FG	100
LW6_VIEW_STYLE	100
LW6_WALL_GREASE	89
LW6_WIDTH	62
LW6_WINDOWED_MODE_LIMIT	62
LW6_X_POLARITY	82
LW6_X_WRAP	100
LW6_Y_POLARITY	83
LW6_Y_WRAP	101
LW6_Z_POLARITY	83
LW6_ZOOM	101
LW6_ZOOM_MAX	101
LW6_ZOOM_MIN	101

M

magic-number	109
map-path	58
max-cursor-pot	71
max-cursor-pot-offset	71
max-map-height	85
max-map-surface	85
max-map-width	86
max-nb-cursors	72
max-nb-nodes	72
max-nb-teams	72
max-round-delta	72
max-zone-size	72
medicine-power	73
memory-bazooka-eraser	110
memory-bazooka-size	110
menu-color-auto	86
menu-color-default-bg	94

menu-color-default-fg	94
menu-color-disabled-bg	95
menu-color-disabled-fg	95
menu-color-selected-bg	95
menu-color-selected-fg	95
menu-style	95
min-map-height	86
min-map-surface	87
min-map-width	87
moves-per-round	73
music-dir	59
music-exclude	96
music-file	96
music-filter	96
music-path	59
music-volume	63

N

nb-attack-tries	73
nb-defense-tries	74
nb-move-tries	74
node-description	65
node-title	65

P

password	65
pilot-lag	110
pixelize	96
public-url	65

R

resample	87
reset-config-on-upgrade	111
respawn-position-mode	74
respawn-team	75
round-delta	75
rounds-per-sec	75

S

side-attack-factor	75
side-defense-factor	76
single-army-size	76
skip-network	66
snd-backend	63
sound-volume	63
speed	87
spread-thread	76
spreads-per-round	76
srv-backends	66
start-blue-x	77
start-blue-y	77
start-cyan-x	77
start-cyan-y	77
start-green-x	78

start-green-y	78
start-lightblue-x	78
start-lightblue-y	78
start-magenta-x	78
start-magenta-y	79
start-orange-x	79
start-orange-y	79
start-pink-x	79
start-pink-y	80
start-position-mode	80
start-purple-x	80
start-purple-y	80
start-red-x	80
start-red-y	81
start-yellow-x	81
start-yellow-y	81
system-color-auto	88
system-color-bg	97
system-color-fg	97

T

target-fps	111
team-color-blue	97
team-color-cyan	97
team-color-dead	97
team-color-green	98
team-color-lightblue	98
team-color-magenta	98
team-color-orange	98
team-color-pink	98
team-color-purple	99
team-color-red	99
team-color-yellow	99
total-armies-size	81
total-time	82

U

upscale-using-bench-value	88
---------------------------------	----

upscale-using-fighter-scale	88
use-hints-xml	67
use-music-file	67
use-rules-xml	67
use-style-xml	67
use-texture	68
user-dir	60

V

vertical-move	82
view-color-auto	89
view-color-cursor-bg	99
view-color-cursor-fg	99
view-color-map-bg	100
view-color-map-fg	100
view-style	100

W

wall-grease	89
width	62
windowed-mode-limit	62

X

x-polarity	82
x-wrap	100

Y

y-polarity	83
y-wrap	101

Z

z-polarity	83
zoom	101
zoom-max	101
zoom-min	101