

# LinuxGL (GLX) QuakeWorld Client compile mini-HOWTO

**Robert B Easter**

**reaster@comptechnews.com**

## Revision History

Revision 1.2 2000-01-04 Revised by: rbe

This miniHOWTO is about how to get the OpenGL/GLX Linux QuakeWorld client compiled and working on systems with standard OpenGL hardware acceleration support.

## 1. Introduction

This miniHOWTO is about how to get the OpenGL/GLX Linux QuakeWorld client compiled and working on systems with standard OpenGL hardware acceleration support.

This procedure was developed while struggling to get glqwcl.glx to work on a system with Linux Slackware 7.0 (2.2.16), XFree86 4.0.1, a GeForce 256 DDR video card, and NVIDIA's 0.9-4 MesaGL drop-in replacement lib/driver (libGL.so.1.0.4).

### 1.1. New Versions of this Document

The latest version of this mini-HOWTO can be found at:

<http://www.linuxdoc.org/HOWTO/mini/LinuxGL-QuakeWorld-mini-HOWTO/>

If you'd like a ready to compile distribution of QuakeWorld that has this HOWTO's procedures performed on it already, then you can download it at:

<http://www.comptechnews.com/~reaster/index.html#games>

## **1.2. Copyright Information**

This document is copyrighted (c) 2000 Robert B. Easter and is distributed under the terms of the Linux Documentation Project (LDP) license, stated below.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have any questions, please contact `<linux-howto@metalab.unc.edu>`

## **1.3. Disclaimer**

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

## **2. QuakeWorld clients**

If you'd like a ready to compile distribution of QuakeWorld that has this HOWTO's procedures performed on it already, then you can download it at:

<http://www.comptechnews.com/~reaster/index.html#games>

Otherwise ...

It seems that the glqwcl.glx binary available at least at [www.quakeworld.net](http://www.quakeworld.net) (<http://www.quakeworld.net/>) in the package `qwcl-2.30-glibc-i386-unknown-linux2.0.tar.gz` is not compiled properly since on startup, it looks for `libglide2x.so`.

QuakeWorld clients come in several flavors:

OpenGL:

glqwcl (3dfx only?)

glqwcl.glx (MesaGL/XFree4)

Software only:

qwcl (svgalib)

qwcl.x11 (x11/svgalib)

- This HOWTO is only concerned with getting the glqwcl.glx working with XFree86 4.x with GLX.
- This HOWTO assumes you have a working OpenGL libGL/libglx installation!

If you happen to have an Nvidia card and are interested in a complete HOWTO for installing the Nvidia drivers and an assortment of OpenGL libraries (GLU, glut, Qt, OpenGL man pages), then see the [NVIDIA-OpenGL-Configuration\\_miniHOWTO](#) at:

<http://www.linuxdoc.org/HOWTO/mini/Nvidia-OpenGL-Configuration-mini-HOWTO/>

### **3. Download the Quake source**

Download the Quake source code, `q1source.zip`, from <ftp://ftp.idsoftware.com/idstuff/source/q1source.zip>. This zip file contains all versions of Quake1/QuakeWorld.

After compiling (read on first!), you will get version:

```
LinuxGL (0.98) QuakeWorld 2.40 (as of this writing, Aug 19, 2000)
```

Make a directory to hold the quake source:

```
mkdir qlsrc
cd qlsrc
unzip ../qlsource.zip
cd QW
cp Makefile.Linux Makefile
```

## 4. Modify the Makefile

The Makefile can use cleaning up and the removal of all Mesa references since the GLX version does not need it. Rather than provide a line by line account of what I did to change the original Makefile, the modified Makefile that I used, and a patch against the original Makefile.Linux to obtain my version of the Makefile, are provided below.

**Figure 1. Makefile, cleaned up for compiling the GLX version.**

```
#
# QuakeWorld Makefile for Linux 2.0
#
# Apr '98 by Zoid <zoid@idsoftware.com>
#
# GNU Make required
#
# ELF only
#

MAINDIR=.

BUILD_RELEASE_DIR=bin
CLIENT_DIR=$(MAINDIR)/client
SERVER_DIR=$(MAINDIR)/server

CC=gcc
BASE_CFLAGS=-Wall -Dstricmp=strcasecmp -I$(CLIENT_DIR) -I$(SERVER_DIR)
RELEASE_CFLAGS=$(BASE_CFLAGS) -m486 -ffast-math -fexpensive-optimizations
GLCFLAGS=-DGLQUAKE -DGL_EXT_SHARED
```

```

LDFLAGS=-lm
XLDFLAGS=-L/usr/X11R6/lib -lX11 -lXext
GL_X11_LDFLAGS=-L/usr/X11R6/lib -lm -lGL -lX11 -lXext

DO_CC=$(CC) $(CFLAGS) -o $@ -c $<
DO_O_CC=$(CC) -O $(CFLAGS) -o $@ -c $<
DO_GL_CC=$(CC) $(CFLAGS) $(GLCFLAGS) -o $@ -c $<
DO_SERVER_CC=$(CC) -DSERVERONLY $(CFLAGS) -o $@ -c $<

DO_AS=$(CC) $(CFLAGS) -DELFB -x assembler-with-cpp -o $@ -c $<
DO_GL_AS=$(CC) $(CFLAGS) $(GLCFLAGS) -DELFB -x assembler-with-cpp -o $@ -c $<

#####
# SETUP AND BUILD
#####

TARGETS=$(BUILDDIR)/qsv $(BUILDDIR)/glqwcl.glx $(BUILDDIR)/qwcl.x11

build_release:
    @-mkdir $(BUILD_RELEASE_DIR) \
        $(BUILD_RELEASE_DIR)/client \
        $(BUILD_RELEASE_DIR)/glclient \
        $(BUILD_RELEASE_DIR)/server
    $(MAKE) targets BUILDDIR=$(BUILD_RELEASE_DIR) CFLAGS="$(RELEASE_CFLAGS)"

all: build_release

targets: $(TARGETS)

#####
# SERVER
#####

QWSV_OBJS = \
    $(BUILDDIR)/server/pr_cmds.o \
    $(BUILDDIR)/server/pr_edict.o \
    $(BUILDDIR)/server/pr_exec.o \
    $(BUILDDIR)/server/sv_init.o \
    $(BUILDDIR)/server/sv_main.o \
    $(BUILDDIR)/server/sv_nchan.o \
    $(BUILDDIR)/server/sv_ents.o \
    $(BUILDDIR)/server/sv_send.o \
    $(BUILDDIR)/server/sv_move.o \
    $(BUILDDIR)/server/sv_phys.o \
    $(BUILDDIR)/server/sv_user.o \
    $(BUILDDIR)/server/sv_ccmds.o \
    $(BUILDDIR)/server/world.o \
    $(BUILDDIR)/server/sys_unix.o \
    $(BUILDDIR)/server/model.o \
    $(BUILDDIR)/server/cmd.o \
    $(BUILDDIR)/server/common.o \
    $(BUILDDIR)/server/crc.o \
    $(BUILDDIR)/server/cvar.o \

```

```
$(BUILDDIR)/server/mathlib.o \  
$(BUILDDIR)/server/md4.o \  
$(BUILDDIR)/server/zone.o \  
$(BUILDDIR)/server/pmove.o \  
$(BUILDDIR)/server/pmovetst.o \  
$(BUILDDIR)/server/net_chan.o \  
$(BUILDDIR)/server/net_udp.o  
  
$(BUILDDIR)/qsv : $(QSV_OBJS)  
$(CC) $(CFLAGS) -o $@ $(QSV_OBJS) $(LDFLAGS)  
  
$(BUILDDIR)/server/pr_cmds.o : $(SERVER_DIR)/pr_cmds.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/pr_edict.o : $(SERVER_DIR)/pr_edict.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/pr_exec.o : $(SERVER_DIR)/pr_exec.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_init.o : $(SERVER_DIR)/sv_init.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_main.o : $(SERVER_DIR)/sv_main.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_nchan.o : $(SERVER_DIR)/sv_nchan.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_ents.o : $(SERVER_DIR)/sv_ents.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_send.o : $(SERVER_DIR)/sv_send.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_move.o : $(SERVER_DIR)/sv_move.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_phys.o : $(SERVER_DIR)/sv_phys.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_user.o : $(SERVER_DIR)/sv_user.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sv_ccmds.o : $(SERVER_DIR)/sv_ccmds.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/world.o : $(SERVER_DIR)/world.c  
$(DO_SERVER_CC)  
  
$(BUILDDIR)/server/sys_unix.o : $(SERVER_DIR)/sys_unix.c  
$(DO_SERVER_CC)
```

```
$(BUILDDIR)/server/model.o : $(SERVER_DIR)/model.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/cmd.o : $(CLIENT_DIR)/cmd.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/common.o : $(CLIENT_DIR)/common.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/crc.o : $(CLIENT_DIR)/crc.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/cvar.o : $(CLIENT_DIR)/cvar.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/mathlib.o : $(CLIENT_DIR)/mathlib.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/md4.o : $(CLIENT_DIR)/md4.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/zone.o : $(CLIENT_DIR)/zone.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/pmove.o : $(CLIENT_DIR)/pmove.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/pmovetst.o : $(CLIENT_DIR)/pmovetst.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/net_chan.o : $(CLIENT_DIR)/net_chan.c
$(DO_SERVER_CC)

$(BUILDDIR)/server/net_udp.o : $(CLIENT_DIR)/net_udp.c
$(DO_SERVER_CC)

#####
# CLIENT
#####

QWCL_OBJS = \
$(BUILDDIR)/client/cl_demo.o \
$(BUILDDIR)/client/cl_ents.o \
$(BUILDDIR)/client/cl_input.o \
$(BUILDDIR)/client/cl_main.o \
$(BUILDDIR)/client/cl_parse.o \
$(BUILDDIR)/client/cl_pred.o \
$(BUILDDIR)/client/cl_tent.o \
$(BUILDDIR)/client/cl_cam.o \
$(BUILDDIR)/client/cmd.o \
$(BUILDDIR)/client/common.o \
$(BUILDDIR)/client/console.o \
$(BUILDDIR)/client/crc.o \
```

```
$(BUILDDIR)/client/cvar.o \  
$(BUILDDIR)/client/d_edge.o \  
$(BUILDDIR)/client/d_fill.o \  
$(BUILDDIR)/client/d_init.o \  
$(BUILDDIR)/client/d_modech.o \  
$(BUILDDIR)/client/d_part.o \  
$(BUILDDIR)/client/d_polyse.o \  
$(BUILDDIR)/client/d_scan.o \  
$(BUILDDIR)/client/d_sky.o \  
$(BUILDDIR)/client/d_sprite.o \  
$(BUILDDIR)/client/d_surf.o \  
$(BUILDDIR)/client/d_vars.o \  
$(BUILDDIR)/client/d_zpoint.o \  
$(BUILDDIR)/client/draw.o \  
$(BUILDDIR)/client/keys.o \  
$(BUILDDIR)/client/mathlib.o \  
$(BUILDDIR)/client/md4.o \  
$(BUILDDIR)/client/menu.o \  
$(BUILDDIR)/client/model.o \  
$(BUILDDIR)/client/net_chan.o \  
$(BUILDDIR)/client/net_udp.o \  
$(BUILDDIR)/client/nonintel.o \  
$(BUILDDIR)/client/pmove.o \  
$(BUILDDIR)/client/pmovetst.o \  
$(BUILDDIR)/client/r_aclip.o \  
$(BUILDDIR)/client/r_alias.o \  
$(BUILDDIR)/client/r_bsp.o \  
$(BUILDDIR)/client/r_draw.o \  
$(BUILDDIR)/client/r_edge.o \  
$(BUILDDIR)/client/r_efrag.o \  
$(BUILDDIR)/client/r_light.o \  
$(BUILDDIR)/client/r_main.o \  
$(BUILDDIR)/client/r_misc.o \  
$(BUILDDIR)/client/r_part.o \  
$(BUILDDIR)/client/r_sky.o \  
$(BUILDDIR)/client/r_sprite.o \  
$(BUILDDIR)/client/r_surf.o \  
$(BUILDDIR)/client/r_vars.o \  
$(BUILDDIR)/client/sbar.o \  
$(BUILDDIR)/client/screen.o \  
$(BUILDDIR)/client/skin.o \  
$(BUILDDIR)/client/snd_dma.o \  
$(BUILDDIR)/client/snd_mem.o \  
$(BUILDDIR)/client/snd_mix.o \  
$(BUILDDIR)/client/view.o \  
$(BUILDDIR)/client/wad.o \  
$(BUILDDIR)/client/zone.o \  
$(BUILDDIR)/client/cd_linux.o \  
$(BUILDDIR)/client/sys_linux.o \  
$(BUILDDIR)/client/snd_linux.o \  
  
QWCL_AS_OBJS = \  
$(BUILDDIR)/client/d_copy.o \  

```



```

$(BUILDDIR)/client/d_draw.o \
$(BUILDDIR)/client/d_draw16.o \
$(BUILDDIR)/client/d_parta.o \
$(BUILDDIR)/client/d_polysa.o \
$(BUILDDIR)/client/d_scana.o \
$(BUILDDIR)/client/d_spr8.o \
$(BUILDDIR)/client/d_varsa.o \
$(BUILDDIR)/client/math.o \
$(BUILDDIR)/client/r_aclipa.o \
$(BUILDDIR)/client/r_aliases.o \
$(BUILDDIR)/client/r_drawa.o \
$(BUILDDIR)/client/r_edgea.o \
$(BUILDDIR)/client/r_varsa.o \
$(BUILDDIR)/client/snd_mixa.o \
$(BUILDDIR)/client/surf16.o \
$(BUILDDIR)/client/surf8.o \
$(BUILDDIR)/client/sys_dosa.o

QWCL_X11_OBJS = $(BUILDDIR)/client/vid_x.o

$(BUILDDIR)/qwcl.x11 : $(QWCL_OBJS) $(QWCL_AS_OBJS) $(QWCL_X11_OBJS)
$(CC) $(CFLAGS) -o $@ $(QWCL_OBJS) $(QWCL_AS_OBJS) $(QWCL_X11_OBJS) \
$(LDFLAGS) $(XLDFLAGS)

$(BUILDDIR)/client/cl_demo.o : $(CLIENT_DIR)/cl_demo.c
$(DO_CC)

$(BUILDDIR)/client/cl_ents.o : $(CLIENT_DIR)/cl_ents.c
$(DO_CC)

$(BUILDDIR)/client/cl_input.o : $(CLIENT_DIR)/cl_input.c
$(DO_CC)

$(BUILDDIR)/client/cl_main.o : $(CLIENT_DIR)/cl_main.c
$(DO_CC)

$(BUILDDIR)/client/cl_parse.o : $(CLIENT_DIR)/cl_parse.c
$(DO_CC)

$(BUILDDIR)/client/cl_pred.o : $(CLIENT_DIR)/cl_pred.c
$(DO_CC)

$(BUILDDIR)/client/cl_tent.o : $(CLIENT_DIR)/cl_tent.c
$(DO_CC)

$(BUILDDIR)/client/cl_cam.o : $(CLIENT_DIR)/cl_cam.c
$(DO_CC)

$(BUILDDIR)/client/cmd.o : $(CLIENT_DIR)/cmd.c
$(DO_CC)

$(BUILDDIR)/client/common.o : $(CLIENT_DIR)/common.c
$(DO_CC)

```

```
$(BUILDDIR)/client/console.o : $(CLIENT_DIR)/console.c
$(DO_CC)

$(BUILDDIR)/client/crc.o : $(CLIENT_DIR)/crc.c
$(DO_CC)

$(BUILDDIR)/client/cvar.o : $(CLIENT_DIR)/cvar.c
$(DO_CC)

$(BUILDDIR)/client/d_edge.o : $(CLIENT_DIR)/d_edge.c
$(DO_CC)

$(BUILDDIR)/client/d_fill.o : $(CLIENT_DIR)/d_fill.c
$(DO_CC)

$(BUILDDIR)/client/d_init.o : $(CLIENT_DIR)/d_init.c
$(DO_CC)

$(BUILDDIR)/client/d_modech.o : $(CLIENT_DIR)/d_modech.c
$(DO_CC)

$(BUILDDIR)/client/d_part.o : $(CLIENT_DIR)/d_part.c
$(DO_CC)

$(BUILDDIR)/client/d_polyse.o : $(CLIENT_DIR)/d_polyse.c
$(DO_CC)

$(BUILDDIR)/client/d_scan.o : $(CLIENT_DIR)/d_scan.c
$(DO_CC)

$(BUILDDIR)/client/d_sky.o : $(CLIENT_DIR)/d_sky.c
$(DO_CC)

$(BUILDDIR)/client/d_sprite.o : $(CLIENT_DIR)/d_sprite.c
$(DO_CC)

$(BUILDDIR)/client/d_surf.o : $(CLIENT_DIR)/d_surf.c
$(DO_CC)

$(BUILDDIR)/client/d_vars.o : $(CLIENT_DIR)/d_vars.c
$(DO_CC)

$(BUILDDIR)/client/d_zpoint.o : $(CLIENT_DIR)/d_zpoint.c
$(DO_CC)

$(BUILDDIR)/client/draw.o : $(CLIENT_DIR)/draw.c
$(DO_CC)

$(BUILDDIR)/client/keys.o : $(CLIENT_DIR)/keys.c
$(DO_CC)

$(BUILDDIR)/client/mathlib.o : $(CLIENT_DIR)/mathlib.c
```

```
$(DO_CC)

$(BUILDDIR)/client/md4.o : $(CLIENT_DIR)/md4.c
$(DO_CC)

$(BUILDDIR)/client/menu.o : $(CLIENT_DIR)/menu.c
$(DO_CC)

$(BUILDDIR)/client/model.o : $(CLIENT_DIR)/model.c
$(DO_CC)

$(BUILDDIR)/client/net_chan.o : $(CLIENT_DIR)/net_chan.c
$(DO_CC)

$(BUILDDIR)/client/net_udp.o : $(CLIENT_DIR)/net_udp.c
$(DO_CC)

$(BUILDDIR)/client/nonintel.o : $(CLIENT_DIR)/nonintel.c
$(DO_CC)

$(BUILDDIR)/client/pmove.o : $(CLIENT_DIR)/pmove.c
$(DO_CC)

$(BUILDDIR)/client/pmovetst.o : $(CLIENT_DIR)/pmovetst.c
$(DO_CC)

$(BUILDDIR)/client/r_aclip.o : $(CLIENT_DIR)/r_aclip.c
$(DO_CC)

$(BUILDDIR)/client/r_alias.o : $(CLIENT_DIR)/r_alias.c
$(DO_CC)

$(BUILDDIR)/client/r_bsp.o : $(CLIENT_DIR)/r_bsp.c
$(DO_CC)

$(BUILDDIR)/client/r_draw.o : $(CLIENT_DIR)/r_draw.c
$(DO_CC)

$(BUILDDIR)/client/r_edge.o : $(CLIENT_DIR)/r_edge.c
$(DO_CC)

$(BUILDDIR)/client/r_efrag.o : $(CLIENT_DIR)/r_efrag.c
$(DO_CC)

$(BUILDDIR)/client/r_light.o : $(CLIENT_DIR)/r_light.c
$(DO_CC)

$(BUILDDIR)/client/r_main.o : $(CLIENT_DIR)/r_main.c
$(DO_CC)

$(BUILDDIR)/client/r_misc.o : $(CLIENT_DIR)/r_misc.c
$(DO_CC)
```

```
$(BUILDDIR)/client/r_part.o : $(CLIENT_DIR)/r_part.c
$(DO_CC)

$(BUILDDIR)/client/r_sky.o : $(CLIENT_DIR)/r_sky.c
$(DO_CC)

$(BUILDDIR)/client/r_sprite.o : $(CLIENT_DIR)/r_sprite.c
$(DO_CC)

$(BUILDDIR)/client/r_surf.o : $(CLIENT_DIR)/r_surf.c
$(DO_CC)

$(BUILDDIR)/client/r_vars.o : $(CLIENT_DIR)/r_vars.c
$(DO_CC)

$(BUILDDIR)/client/sbar.o : $(CLIENT_DIR)/sbar.c
$(DO_CC)

$(BUILDDIR)/client/screen.o : $(CLIENT_DIR)/screen.c
$(DO_CC)

$(BUILDDIR)/client/skin.o : $(CLIENT_DIR)/skin.c
$(DO_CC)

$(BUILDDIR)/client/snd_dma.o : $(CLIENT_DIR)/snd_dma.c
$(DO_CC)

$(BUILDDIR)/client/snd_mem.o : $(CLIENT_DIR)/snd_mem.c
$(DO_CC)

$(BUILDDIR)/client/snd_mix.o : $(CLIENT_DIR)/snd_mix.c
$(DO_CC)

$(BUILDDIR)/client/view.o : $(CLIENT_DIR)/view.c
$(DO_CC)

$(BUILDDIR)/client/wad.o : $(CLIENT_DIR)/wad.c
$(DO_CC)

$(BUILDDIR)/client/zone.o : $(CLIENT_DIR)/zone.c
$(DO_CC)

$(BUILDDIR)/client/cd_linux.o : $(CLIENT_DIR)/cd_linux.c
$(DO_CC)

$(BUILDDIR)/client/sys_linux.o : $(CLIENT_DIR)/sys_linux.c
$(DO_CC)

$(BUILDDIR)/client/snd_linux.o : $(CLIENT_DIR)/snd_linux.c
$(DO_CC)

$(BUILDDIR)/client/d_copy.o : $(CLIENT_DIR)/d_copy.s
$(DO_AS)
```

```
$(BUILDDIR)/client/d_draw.o : $(CLIENT_DIR)/d_draw.s
$(DO_AS)

$(BUILDDIR)/client/d_draw16.o : $(CLIENT_DIR)/d_draw16.s
$(DO_AS)

$(BUILDDIR)/client/d_parta.o : $(CLIENT_DIR)/d_parta.s
$(DO_AS)

$(BUILDDIR)/client/d_polysa.o : $(CLIENT_DIR)/d_polysa.s
$(DO_AS)

$(BUILDDIR)/client/d_scana.o : $(CLIENT_DIR)/d_scana.s
$(DO_AS)

$(BUILDDIR)/client/d_spr8.o : $(CLIENT_DIR)/d_spr8.s
$(DO_AS)

$(BUILDDIR)/client/d_varsa.o : $(CLIENT_DIR)/d_varsa.s
$(DO_AS)

$(BUILDDIR)/client/math.o : $(CLIENT_DIR)/math.s
$(DO_AS)

$(BUILDDIR)/client/r_aclipa.o : $(CLIENT_DIR)/r_aclipa.s
$(DO_AS)

$(BUILDDIR)/client/r_aliasa.o : $(CLIENT_DIR)/r_aliasa.s
$(DO_AS)

$(BUILDDIR)/client/r_drawa.o : $(CLIENT_DIR)/r_drawa.s
$(DO_AS)

$(BUILDDIR)/client/r_edgea.o : $(CLIENT_DIR)/r_edgea.s
$(DO_AS)

$(BUILDDIR)/client/r_varsa.o : $(CLIENT_DIR)/r_varsa.s
$(DO_AS)

$(BUILDDIR)/client/snd_mixa.o : $(CLIENT_DIR)/snd_mixa.s
$(DO_AS)

$(BUILDDIR)/client/surf16.o : $(CLIENT_DIR)/surf16.s
$(DO_AS)

$(BUILDDIR)/client/surf8.o : $(CLIENT_DIR)/surf8.s
$(DO_AS)

$(BUILDDIR)/client/sys_dosa.o : $(CLIENT_DIR)/sys_dosa.s
$(DO_AS)

$(BUILDDIR)/client/vid_x.o : $(CLIENT_DIR)/vid_x.c
```

```

$(DO_CC)

#####
# GL CLIENT
#####

GLQWCL_OBJS = \
$(BUILDDIR)/glclient/cl_demo.o \
$(BUILDDIR)/glclient/cl_ents.o \
$(BUILDDIR)/glclient/cl_input.o \
$(BUILDDIR)/glclient/cl_main.o \
$(BUILDDIR)/glclient/cl_parse.o \
$(BUILDDIR)/glclient/cl_pred.o \
$(BUILDDIR)/glclient/cl_tent.o \
$(BUILDDIR)/glclient/cl_cam.o \
$(BUILDDIR)/glclient/cmd.o \
$(BUILDDIR)/glclient/common.o \
$(BUILDDIR)/glclient/console.o \
$(BUILDDIR)/glclient/crc.o \
$(BUILDDIR)/glclient/cvar.o \
$(BUILDDIR)/glclient/keys.o \
$(BUILDDIR)/glclient/mathlib.o \
$(BUILDDIR)/glclient/md4.o \
$(BUILDDIR)/glclient/menu.o \
$(BUILDDIR)/glclient/net_chan.o \
$(BUILDDIR)/glclient/net_udp.o \
$(BUILDDIR)/glclient/nonintel.o \
$(BUILDDIR)/glclient/pmove.o \
$(BUILDDIR)/glclient/pmovetst.o \
$(BUILDDIR)/glclient/r_part.o \
$(BUILDDIR)/glclient/sbar.o \
$(BUILDDIR)/glclient/skin.o \
$(BUILDDIR)/glclient/snd_dma.o \
$(BUILDDIR)/glclient/snd_mem.o \
$(BUILDDIR)/glclient/snd_mix.o \
$(BUILDDIR)/glclient/view.o \
$(BUILDDIR)/glclient/wad.o \
$(BUILDDIR)/glclient/zone.o \
$(BUILDDIR)/glclient/cd_linux.o \
$(BUILDDIR)/glclient/sys_linux.o \
$(BUILDDIR)/glclient/snd_linux.o \
\
$(BUILDDIR)/glclient/gl_draw.o \
$(BUILDDIR)/glclient/gl_mesh.o \
$(BUILDDIR)/glclient/gl_model.o \
$(BUILDDIR)/glclient/gl_ngraph.o \
$(BUILDDIR)/glclient/gl_refrag.o \
$(BUILDDIR)/glclient/gl_rlight.o \
$(BUILDDIR)/glclient/gl_rmain.o \
$(BUILDDIR)/glclient/gl_rmisc.o \
$(BUILDDIR)/glclient/gl_rsurf.o \
$(BUILDDIR)/glclient/gl_screen.o \
$(BUILDDIR)/glclient/gl_warp.o \

```

```

\
$(BUILDDIR)/glclient/math.o \
$(BUILDDIR)/glclient/snd_mixa.o \
$(BUILDDIR)/glclient/sys_dosa.o

GLQWCL_X11_OBJS = $(BUILDDIR)/glclient/gl_vidlinuxglx.o

$(BUILDDIR)/glqwcl.glx : $(GLQWCL_OBJS) $(GLQWCL_X11_OBJS)
$(CC) $(CFLAGS) -o $@ $(GLQWCL_OBJS) $(GLQWCL_X11_OBJS) $(LDFLAGS) $(GL_X11_LDFLAGS)

$(BUILDDIR)/glclient/cl_demo.o : $(CLIENT_DIR)/cl_demo.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cl_ents.o : $(CLIENT_DIR)/cl_ents.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cl_input.o : $(CLIENT_DIR)/cl_input.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cl_main.o : $(CLIENT_DIR)/cl_main.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cl_parse.o : $(CLIENT_DIR)/cl_parse.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cl_pred.o : $(CLIENT_DIR)/cl_pred.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cl_tent.o : $(CLIENT_DIR)/cl_tent.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cl_cam.o : $(CLIENT_DIR)/cl_cam.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cmd.o : $(CLIENT_DIR)/cmd.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/common.o : $(CLIENT_DIR)/common.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/console.o : $(CLIENT_DIR)/console.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/crc.o : $(CLIENT_DIR)/crc.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/cvar.o : $(CLIENT_DIR)/cvar.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/keys.o : $(CLIENT_DIR)/keys.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/mathlib.o : $(CLIENT_DIR)/mathlib.c

```

```
$(DO_GL_CC)

$(BUILDDIR)/glclient/md4.o : $(CLIENT_DIR)/md4.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/menu.o : $(CLIENT_DIR)/menu.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/net_chan.o : $(CLIENT_DIR)/net_chan.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/net_udp.o : $(CLIENT_DIR)/net_udp.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/nonintel.o : $(CLIENT_DIR)/nonintel.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/pmove.o : $(CLIENT_DIR)/pmove.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/pmovetst.o : $(CLIENT_DIR)/pmovetst.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/r_part.o : $(CLIENT_DIR)/r_part.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/sbar.o : $(CLIENT_DIR)/sbar.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/screen.o : $(CLIENT_DIR)/screen.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/skin.o : $(CLIENT_DIR)/skin.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/snd_dma.o : $(CLIENT_DIR)/snd_dma.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/snd_mem.o : $(CLIENT_DIR)/snd_mem.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/snd_mix.o : $(CLIENT_DIR)/snd_mix.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/view.o : $(CLIENT_DIR)/view.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/wad.o : $(CLIENT_DIR)/wad.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/zone.o : $(CLIENT_DIR)/zone.c
$(DO_GL_CC)
```



```
$(BUILDDIR)/glclient/cd_linux.o : $(CLIENT_DIR)/cd_linux.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/sys_linux.o : $(CLIENT_DIR)/sys_linux.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/snd_linux.o : $(CLIENT_DIR)/snd_linux.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_draw.o : $(CLIENT_DIR)/gl_draw.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_mesh.o : $(CLIENT_DIR)/gl_mesh.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_model.o : $(CLIENT_DIR)/gl_model.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_ngraph.o : $(CLIENT_DIR)/gl_ngraph.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_refrag.o : $(CLIENT_DIR)/gl_refrag.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_rlight.o : $(CLIENT_DIR)/gl_rlight.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_rmain.o : $(CLIENT_DIR)/gl_rmain.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_rmisc.o : $(CLIENT_DIR)/gl_rmisc.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_rsurf.o : $(CLIENT_DIR)/gl_rsurf.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_screen.o : $(CLIENT_DIR)/gl_screen.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_vidlinux.o : $(CLIENT_DIR)/gl_vidlinux.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_vidlinuxglx.o : $(CLIENT_DIR)/gl_vidlinuxglx.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/gl_warp.o : $(CLIENT_DIR)/gl_warp.c
$(DO_GL_CC)

$(BUILDDIR)/glclient/math.o : $(CLIENT_DIR)/math.s
$(DO_GL_AS)

$(BUILDDIR)/glclient/snd_mixa.o : $(CLIENT_DIR)/snd_mixa.s
$(DO_GL_AS)
```

```
$(BUILDDIR)/glclient/sys_dosa.o :      $(CLIENT_DIR)/sys_dosa.s
$(DO_GL_AS)
```

```
#####
# MISC
#####

clean:
rm -fr bin/client
rm -fr bin/glclient
rm -fr bin/server
```

This file can be copied and pasted as your Makefile to replace the original Makefile/Makefile.Linux.

The following is a patch that you can apply to the original Makefile.Linux if you feel like it. The command is simply:

```
patch Makefile.Linux Makefile.Linux.patch
cp Makefile.Linux Makefile
```

You can download it at:

<http://www.comptechnews.com/~reaster/mini-HOWTO/LinuxGL-QuakeWorld-mini-HOWTO/Makefile.Linux.patch>

### Figure 2. Makefile.Linux.patch

```
*** /home/reaster/q1source/QW/Makefile.Linux Wed Apr 21 20:16:24 1999
--- Makefile Tue Nov 28 09:36:47 2000
*****
*** 8,59 ****
# ELF only
#

! # RPM release number
! RPM_RELEASE=1

! VERSION=2.30$(GLIBC)
!
! ifneq (,$(findstring libc6,$(shell if [ -e /lib/libc.so.6* ];then echo libc6;fi)))
! GLIBC=-glibc
! else
! GLIBC=
! endif
!
! ifneq (,$(findstring alpha,$(shell uname -m)))
```

```

! ARCH=axp
! RPMARCH=alpha
! else
! ARCH=i386
! RPMARCH=i386
! endif
!
! MAINDIR=/grog/Projects/QW
!
! BUILD_DEBUG_DIR=debug$(ARCH)$(GLIBC)
! BUILD_RELEASE_DIR=release$(ARCH)$(GLIBC)
! CLIENT_DIR=$(MAINDIR)/client
! SERVER_DIR=$(MAINDIR)/server

- MESA_DIR=/usr/local/src/Mesa-3.0
-
! CC=gcc
! BASE_CFLAGS=-Wall -Dstricmp=strcasecmp -I$(CLIENT_DIR) -I$(SERVER_DIR)
! DEBUG_CFLAGS=$(BASE_CFLAGS) -g
! ifeq ($(ARCH),axp)
! RELEASE_CFLAGS=$(BASE_CFLAGS) -ffast-math -funroll-loops \
! -fomit-frame-pointer -fexpensive-optimizations
! else
! RELEASE_CFLAGS=$(BASE_CFLAGS) -m486 -O6 -ffast-math -funroll-loops \
! -fomit-frame-pointer -fexpensive-optimizations -malign-loops=2 \
! -malign-jumps=2 -malign-functions=2
! endif
! GLCFLAGS=-DGLQUAKE -I/usr/local/src/Mesa-3.0/include -I/usr/include/glide

! LDFLAGS=-lm
- SVGALDFLAGS=-lvga
! XLDFLAGS=-L/usr/X11R6/lib -lX11 -lXext
! GL_SVGA_LDFLAGS=-L/usr/X11/lib -L/usr/local/src/Mesa-3.0/lib -lm -lMesaGL -lglide2x -lX11
! GL_X11_LDFLAGS=-L/usr/X11/lib -L/usr/local/src/Mesa-3.0/lib -lm -lGL -lX11 -lXext

! DO_CC=$(CC) $(CFLAGS) -o $@ -c $<
! DO_O_CC=$(CC) -O $(CFLAGS) -o $@ -c $<
--- 8,27 ----
! # ELF only
! #

! MAINDIR=.

! BUILD_RELEASE_DIR=bin
! CLIENT_DIR=$(MAINDIR)/client
! SERVER_DIR=$(MAINDIR)/server

! CC=gcc
! BASE_CFLAGS=-Wall -Dstricmp=strcasecmp -I$(CLIENT_DIR) -I$(SERVER_DIR)
! RELEASE_CFLAGS=$(BASE_CFLAGS) -m486 -ffast-math -fexpensive-optimizations
! GLCFLAGS=-DGLQUAKE -DGL_EXT_SHARED

! LDFLAGS=-lm

```

```

XLDLFLAGS=-L/usr/X11R6/lib -lX11 -lXext
! GL_X11_LDFLAGS=-L/usr/X11R6/lib -lm -lGL -lX11 -lXext

DO_CC=$(CC) $(CFLAGS) -o $@ -c $<
DO_O_CC=$(CC) -O $(CFLAGS) -o $@ -c $<
*****
*** 67,85 ****
# SETUP AND BUILD
#####

! ifeq ($(ARCH),x86)
! TARGETS=$(BUILDDIR)/qwsv
! #$(BUILDDIR)/qwcl.x11
! else
! TARGETS=$(BUILDDIR)/qwsv $(BUILDDIR)/qwcl $(BUILDDIR)/qwcl.x11 $(BUILDDIR)/glqwcl $(BUILDDIR)/glqwcl.x11
! endif
!
! build_debug:
! @-mkdir $(BUILD_DEBUG_DIR) \
! $(BUILD_DEBUG_DIR)/client \
! $(BUILD_DEBUG_DIR)/glclient \
! $(BUILD_DEBUG_DIR)/server
! $(MAKE) targets BUILDDIR=$(BUILD_DEBUG_DIR) CFLAGS="$(DEBUG_CFLAGS)"

build_release:
@-mkdir $(BUILD_RELEASE_DIR) \
--- 35,41 ----
# SETUP AND BUILD
#####

! TARGETS=$(BUILDDIR)/qwsv $(BUILDDIR)/glqwcl.glx $(BUILDDIR)/qwcl.x11

build_release:
@-mkdir $(BUILD_RELEASE_DIR) \
*****
*** 88,94 ****
$(BUILD_RELEASE_DIR)/server
$(MAKE) targets BUILDDIR=$(BUILD_RELEASE_DIR) CFLAGS="$(RELEASE_CFLAGS)"

! all: build_debug build_release

targets: $(TARGETS)

--- 44,50 ----
$(BUILD_RELEASE_DIR)/server
$(MAKE) targets BUILDDIR=$(BUILD_RELEASE_DIR) CFLAGS="$(RELEASE_CFLAGS)"

! all: build_release

targets: $(TARGETS)

*****
*** 273,279 ****

```

```

$(BUILDDIR)/client/sys_linux.o \
$(BUILDDIR)/client/snd_linux.o \

- ifeq ($(ARCH),i386)
  QWCL_AS_OBJS = \
  $(BUILDDIR)/client/d_copy.o \
  $(BUILDDIR)/client/d_draw.o \
--- 229,234 ----
*****
*** 293,309 ****
  $(BUILDDIR)/client/surfl6.o \
  $(BUILDDIR)/client/surf8.o \
  $(BUILDDIR)/client/sys_dosa.o
- else
- QWCL_AS_OBJS=
- endif

- QWCL_SVGA_OBJS = $(BUILDDIR)/client/vid_svgalib.o
  QWCL_X11_OBJS = $(BUILDDIR)/client/vid_x.o

- $(BUILDDIR)/qwcl : $(QWCL_OBJS) $(QWCL_AS_OBJS) $(QWCL_SVGA_OBJS)
- $(CC) $(CFLAGS) -o $@ $(QWCL_OBJS) $(QWCL_AS_OBJS) $(QWCL_SVGA_OBJS) \
- $(LDFLAGS) $(SVGALDFLAGS)
-
  $(BUILDDIR)/qwcl.x11 : $(QWCL_OBJS) $(QWCL_AS_OBJS) $(QWCL_X11_OBJS)
  $(CC) $(CFLAGS) -o $@ $(QWCL_OBJS) $(QWCL_AS_OBJS) $(QWCL_X11_OBJS) \
  $(LDFLAGS) $(XLDFLAGS)
--- 248,256 ----
*****
*** 548,556 ****
  $(BUILDDIR)/client/sys_dosa.o :      $(CLIENT_DIR)/sys_dosa.s
  $(DO_AS)

- $(BUILDDIR)/client/vid_svgalib.o : $(CLIENT_DIR)/vid_svgalib.c
- $(DO_O_CC)
-
  $(BUILDDIR)/client/vid_x.o : $(CLIENT_DIR)/vid_x.c
  $(DO_CC)

--- 495,500 ----
*****
*** 610,621 ****
  $(BUILDDIR)/glclient/snd_mixa.o \
  $(BUILDDIR)/glclient/sys_dosa.o

- GLQWCL_SVGA_OBJS = $(BUILDDIR)/glclient/gl_vidlinux.o
  GLQWCL_X11_OBJS = $(BUILDDIR)/glclient/gl_vidlinuxglx.o

- $(BUILDDIR)/glqwcl : $(GLQWCL_OBJS) $(GLQWCL_SVGA_OBJS)
- $(CC) $(CFLAGS) -o $@ $(GLQWCL_OBJS) $(GLQWCL_SVGA_OBJS) $(LDFLAGS) $(GL_SVGA_LDFLAGS)
-
  $(BUILDDIR)/glqwcl.glx : $(GLQWCL_OBJS) $(GLQWCL_X11_OBJS)
  $(CC) $(CFLAGS) -o $@ $(GLQWCL_OBJS) $(GLQWCL_X11_OBJS) $(LDFLAGS) $(GL_X11_LDFLAGS)

```

```

--- 554,561 ----
*****
*** 772,930 ***
  $(BUILDDIR)/glclient/sys_dosa.o :      $(CLIENT_DIR)/sys_dosa.s
  $(DO_GL_AS)

- #####
- # RPM
- #####
-
- # Make RPMs.  You need to be root to make this work
- RPMROOT=/usr/src/redhat
- RPM = rpm
- RPMFLAGS = -bb
- INSTALLDIR = /usr/local/games/quake
-
- ifeq ($(ARCH),axp)
- rpm: rpm-qwsv
- tar: tar-qwsv
- else
- rpm: rpm-qwcl rpm-qwsv
- tar: tar-qwcl tar-qwsv
- endif
-
- QWCL_RPMDIR=/var/tmp/qwcl-$(VERSION)
- TDFXGL_DIR=/home/zoid/3dfxgl
-
- QWPROGS = \
- $(MAINDIR)/progs/buttons.qc \
- $(MAINDIR)/progs/client.qc \
- $(MAINDIR)/progs/combat.qc \
- $(MAINDIR)/progs/defs.qc \
- $(MAINDIR)/progs/doors.qc \
- $(MAINDIR)/progs/items.qc \
- $(MAINDIR)/progs/misc.qc \
- $(MAINDIR)/progs/models.qc \
- $(MAINDIR)/progs/plats.qc \
- $(MAINDIR)/progs/player.qc \
- $(MAINDIR)/progs/progdefs.h \
- $(MAINDIR)/progs/progs.src \
- $(MAINDIR)/progs/qwprogs.dat \
- $(MAINDIR)/progs/server.qc \
- $(MAINDIR)/progs/spectate.qc \
- $(MAINDIR)/progs/sprites.qc \
- $(MAINDIR)/progs/subs.qc \
- $(MAINDIR)/progs/triggers.qc \
- $(MAINDIR)/progs/weapons.qc \
- $(MAINDIR)/progs/world.qc
-
- rpm-qwcl: qwcl.spec \
- $(BUILD_RELEASE_DIR)/qwcl \
- $(BUILD_RELEASE_DIR)/glqwcl \

```

```

- $(BUILD_RELEASE_DIR)/glqwcl.glx \
- $(BUILD_RELEASE_DIR)/qwcl.x11
- touch $(RPMROOT)/SOURCES/qwcl-$(VERSION).tar.gz
- if [ ! -d archives ];then mkdir archives;fi
- $(MAKE) copyfiles-qwcl DESTDIR=$(QWCL_RPMDIR)/$(INSTALLDIR)
- cp qwcl.spec $(RPMROOT)/SPECS/qwcl.spec
- cp $(MAINDIR)/quake.gif $(RPMROOT)/SOURCES/quake.gif
- cd $(RPMROOT)/SPECS; $(RPM) $(RPMFLAGS) qwcl.spec
- cp $(RPMROOT)/RPMS/$(RPMARCH)/qwcl-$(VERSION)-$(RPM_RELEASE).$(RPMARCH).rpm archives/.
- rm -rf $(QWCL_RPMDIR)
-
- tar-qwcl: $(BUILD_RELEASE_DIR)/qwcl \
- $(BUILD_RELEASE_DIR)/glqwcl \
- $(BUILD_RELEASE_DIR)/glqwcl.glx \
- $(BUILD_RELEASE_DIR)/qwcl.x11
- if [ ! -d archives ];then mkdir archives;fi
- $(MAKE) copyfiles-qwcl DESTDIR=$(QWCL_RPMDIR)/$(INSTALLDIR)
- cd $(QWCL_RPMDIR)/$(INSTALLDIR); tar czvf qwcl-$(VERSION)-$(RPMARCH)-unknown-linux2.0.ta
- mv $(QWCL_RPMDIR)/$(INSTALLDIR)/*.tar.gz archives/.
- rm -rf $(QWCL_RPMDIR)
-
- copyfiles-qwcl:
- -mkdirhier $(DESTDIR)
- -mkdirhier $(DESTDIR)/qw
- -mkdirhier $(DESTDIR)/qw/skins
- cp $(BUILD_RELEASE_DIR)/qwcl $(DESTDIR)/.
- strip $(DESTDIR)/qwcl
- chmod 4755 $(DESTDIR)/qwcl
- cp $(BUILD_RELEASE_DIR)/qwcl.x11 $(DESTDIR)/.
- strip $(DESTDIR)/qwcl.x11
- chmod 755 $(DESTDIR)/qwcl.x11
- cp $(BUILD_RELEASE_DIR)/glqwcl $(DESTDIR)/.
- strip $(DESTDIR)/glqwcl
- chmod 4755 $(DESTDIR)/glqwcl
- cp $(BUILD_RELEASE_DIR)/glqwcl.glx $(DESTDIR)/.
- strip $(DESTDIR)/glqwcl.glx
- chmod 4755 $(DESTDIR)/glqwcl.glx
- cp $(TDFXGL_DIR)/release$(ARCH)/lib3dfxgl.so $(DESTDIR)/.
- chmod 755 $(DESTDIR)/lib3dfxgl.so
- cp $(MAINDIR)/glqwcl.3dfxgl $(DESTDIR)/.
- chmod 755 $(DESTDIR)/glqwcl.3dfxgl
- cp $(MAINDIR)/docs/README.qwcl $(DESTDIR)/.
- chmod 644 $(DESTDIR)/README.qwcl
- cp $(MAINDIR)/fixskins.sh $(DESTDIR)/qw/skins/.
- chmod 755 $(DESTDIR)/qw/skins/fixskins.sh
- cp $(MESA_DIR)/lib/libMesaGL.so.2.6 $(DESTDIR)
- chmod 755 $(DESTDIR)/libMesaGL.so.2.6
- cd $(DESTDIR); ln -sf libMesaGL.so.2.6 libMesaGL.so.2
- cd $(DESTDIR); ln -sf libMesaGL.so.2 libMesaGL.so
- cd $(DESTDIR); ln -sf libMesaGL.so libGL.so
-
- qwcl.spec : $(MAINDIR)/qwcl.spec.sh $(BUILD_RELEASE_DIR)/qwcl
- sh $< $(VERSION) $(RPM_RELEASE) $(INSTALLDIR) > $@

```

```

-
- QWSV_RPMDIR=/var/tmp/qwsv-$(VERSION)
- rpm-qwsv: qwsv.spec $(BUILD_RELEASE_DIR)/qwsv $(QWPROGS)
- touch $(RPMROOT)/SOURCES/qwsv-$(VERSION).tar.gz
- if [ ! -d archives ];then mkdir archives;fi
- $(MAKE) copyfiles-qwsv DESTDIR=$(QWSV_RPMDIR)/$(INSTALLDIR)
- cp qwsv.spec $(RPMROOT)/SPECS/qwsv.spec
- cp $(MAINDIR)/quake.gif $(RPMROOT)/SOURCES/quake.gif
- cd $(RPMROOT)/SPECS; $(RPM) $(RPMFLAGS) qwsv.spec
- cp $(RPMROOT)/RPM/$ (RPMARCH)/qwsv-$(VERSION)-$(RPM_RELEASE).$(RPMARCH).rpm archives/.
- rm -rf $(QWSV_RPMDIR)
-
- tar-qwsv: $(BUILD_RELEASE_DIR)/qwsv $(QWPROGS)
- if [ ! -d archives ];then mkdir archives;fi
- $(MAKE) copyfiles-qwsv DESTDIR=$(QWSV_RPMDIR)/$(INSTALLDIR)
- cd $(QWSV_RPMDIR)/$(INSTALLDIR); tar czvf qwsv-$(VERSION)-$(RPMARCH)-unknown-linux2.0.ta
- mv $(QWSV_RPMDIR)/$(INSTALLDIR)/*.tar.gz archives/.
- rm -rf $(QWSV_RPMDIR)
-
- copyfiles-qwsv:
- -mkdirhier $(DESTDIR)
- -mkdirhier $(DESTDIR)/qw
- -mkdirhier $(DESTDIR)/qw/skins
- cp $(BUILD_RELEASE_DIR)/qwsv $(DESTDIR)/.
- strip $(DESTDIR)/qwsv
- chmod 755 $(DESTDIR)/qwsv
- cp $(MAINDIR)/docs/README.qwsv $(DESTDIR)/.
- chmod 644 $(DESTDIR)/README.qwsv
- cp $(QWPROGS) $(DESTDIR)/qw/.
- cd $(DESTDIR)/qw; chmod 644 *
- chmod 755 $(DESTDIR)/qw/skins
- cp $(MAINDIR)/fixskins.sh $(DESTDIR)/qw/skins/.
- chmod 755 $(DESTDIR)/qw/skins/fixskins.sh
-
- qwsv.spec : $(MAINDIR)/qwsv.spec.sh $(BUILD_RELEASE_DIR)/qwsv
- sh $< $(VERSION) $(RPM_RELEASE) $(INSTALLDIR) > $@

#####
# MISC
#####

! clean: clean-debug clean-release
!
! clean-debug:
! $(MAKE) clean2 BUILDDIR=$(BUILD_DEBUG_DIR) CFLAGS="$(DEBUG_CFLAGS)"
!
! clean-release:
! $(MAKE) clean2 BUILDDIR=$(BUILD_RELEASE_DIR) CFLAGS="$(DEBUG_CFLAGS)"

- clean2:
- -rm -f $(QWSV_OBJS) \
- $(QWCL_OBJS) \
- $(QWCL_AS_OBJS) \

```



```

- $(QWCL_SVGA_OBJS) \
- $(QWCL_X11_OBJS) \
- $(GLQWCL_OBJS) \

--- 712,725 ----
$(BUILDDIR)/glclient/sys_dosa.o :      $(CLIENT_DIR)/sys_dosa.s
$(DO_GL_AS)

#####
# MISC
#####

! clean:
! rm -fr bin/client
! rm -fr bin/glclient
! rm -fr bin/server

```

Save the Makefile.

## 5. Fix the mouse grabbing

Fix the mouse hide problem.

The quake code, as it is, does not hide the mouse when you use it. The pointer would remain visible and jitter around a fixed point in the middle of the screen - very annoying. I found the following simple fix in a mail archive about glx development at:

<http://lists.openprojects.net/pipermail/glx-dev/1999-October/000994.html>

See the patch listing and the link to download it below, or you can make the changes yourself.

```

cd qlsrc/QW/client
edit the file: gl_vidlinuxglx.c

```

At the top of the function called `install_grabs(void)` (Line 234), add the following code:

```

static void install_grabs(void)
{
    /* vars to make blank cursor */
    Pixmap blank;
    XColor dummy;

```

```

char data[1] = {0};
Cursor cursor;

/* make a blank cursor */
blank = XCreateBitmapFromData (dpy, win, data, 1, 1);
if(blank == None) fprintf(stderr, "error: out of memory.\n");
cursor = XCreatePixmapCursor(dpy, blank, blank, &dummy, &dummy, 0, 0);
XFreePixmap (dpy, blank);
Con_Printf("Cursor blanked.\n");

```

The 8th parameter to XGrabPointer should be changed from "None" to "cursor" to use the blank cursor:

```

XGrabPointer(dpy, win,
             True,
             0,
             GrabModeAsync, GrabModeAsync,
             win,
             cursor,
             CurrentTime);

```

(see: man XGrabPointer)

This fix might work for the other client builds if you make similar changes in the appropriate \*vid\*.c file. As it is, this creates a memory leak as it allocates another Cursor object everytime through and doesn't call XFreeCursor() but this is probably not a problem since I don't think a 1x1 cursor is using much memory. If you want, you can apply the following patch against `q1src/QW/client/gl_vidlinuxglx.c` (**patch `gl_vidlinuxglx.c` patchfile**). Just cut and paste the contents between the begin/end patchfile lines into a file called `gl_vidlinuxglx.c.patch` (or whatever you want to call it). Save patchfile in the `q1src/QW/client` directory where `gl_vidlinuxglx.c` is. Then run the command:

```
patch gl_vidlinuxglx.c gl_vidlinuxglx.c.patch
```

This patch file will make `gl_vidlinuxglx.c` such that it doesn't have any memory leak. You can download this patch file at: [http://www.comptechnews.com/~reaster/mini-HOWTO/LinuxGL-QuakeWorld-mini-HOWTO/gl\\_vidlinuxglx.c.patch](http://www.comptechnews.com/~reaster/mini-HOWTO/LinuxGL-QuakeWorld-mini-HOWTO/gl_vidlinuxglx.c.patch).

```

***** begin patchfile *****
*** q1src-p/QW/client/gl_vidlinuxglx.c Tue Dec 21 18:45:54 1999
--- gl_vidlinuxglx.c Sat Aug 19 20:47:42 2000
*****
*** 1,22 ****
! /*

```

```
! Copyright (C) 1996-1997 Id Software, Inc.
!
! 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 2
! 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, write to the Free Software
! Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
!
! */
#include <termios.h>
#include <sys/ioctl.h>
#include <sys/stat.h>
--- 1,22 ----
! /*
! Copyright (C) 1996-1997 Id Software, Inc.
!
! 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 2
! 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, write to the Free Software
! Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
!
! */
#include <termios.h>
#include <sys/ioctl.h>
#include <sys/stat.h>
*****
*** 66,71 ****
--- 66,74 ----

static int scr_width, scr_height;

+ /* blank cursor - gets initialized in VID_Init, freed in VID_Shutdown */
+ Cursor cursor;
+
```

```

/*-----*/

//int texture_mode = GL_NEAREST;
*****
*** 236,243 ****
    0,
    GrabModeAsync, GrabModeAsync,
    win,
!   None,
    CurrentTime);

#ifdef USE_DGA
    XF86DGADirectVideo(dpy, DefaultScreen(dpy), XF86DGADirectMouse);
--- 239,247 ----
    0,
    GrabModeAsync, GrabModeAsync,
    win,
!   cursor,
    CurrentTime);
+ Con_Printf("Cursor blanked.\n");

#ifdef USE_DGA
    XF86DGADirectVideo(dpy, DefaultScreen(dpy), XF86DGADirectMouse);
*****
*** 350,355 ****
--- 354,360 ----
    if (!ctx)
        return;

+ XFreeCursor(dpy, cursor);
  glXDestroyContext(dpy, ctx);
}

*****
*** 602,608 ****
    unsigned long mask;
    Window root;
    XVisualInfo *visinfo;
!
    S_Init();

    Cvar_RegisterVariable (&vid_mode);
--- 607,617 ----
    unsigned long mask;
    Window root;
    XVisualInfo *visinfo;
! /* vars to make blank cursor */
! Pixmap blank;
! XColor dummy;
! char data[1] = {0};
!
    S_Init();

```

```

    Cvar_RegisterVariable (&vid_mode);
*****
*** 701,706 ***
--- 710,720 ----
    Con_SafePrintf ("Video mode %dx%d initialized.\n", width, height);

    vid.recalc_refdef = 1;    // force a surface cache flush
+
+ blank = XCreateBitmapFromData(dpy, win, data, 1, 1);
+ if(blank == None) fprintf(stderr, "error: out of memory.\n");
+ cursor = XCreatePixmapCursor(dpy, blank, blank, &dummy, &dummy, 0, 0);
+ XFreePixmap(dpy, blank);
+ }

    void Sys_SendKeyEvents(void)

***** end patchfile *****

```

It is handy to bind some keys at the quake console to toggle mouse grabbing:

```

bind q "_windowed_mouse 0"
bind w "_windowed_mouse 1"

```

## 6. Compile!

Compile (finally!)

Just run `make` with no parameters from where the Makefile is located:

```

cd qlsrc/QW
make

```

If all goes well, you will have a binary, `glqwcl.glx`, in the `releasei386-glibc` directory. Copy `glqwcl.glx` to another directory (like "qw") and then from a Quake cd or other quake installation, copy the `id1` directory and any other game directories you like/have - `ctf`, `Fortress` etc - to your new `quakeworld` directory.

## 7. Running glqwcl.glx

Running the game.

You might want to run it like this from XQF: **glqwcl.glx -width 1280 -height 1024** (or any other resolution your system can support)

XQF is a nice program to find QuakeWorld servers and can be found at: <http://www.linuxgames.com/xqf>

To run the game fullscreen requires starting the game at the same resolution as your X desktop. If you are running a desktop environment, you might not be able to hide everything. The sure way to get a fullscreen experience is to run XFree86 on its own without any desktop environment or window manager loaded:

### **XFree86 &**

This starts X without any desktop environment or window manager. `/etc/X11/xinit/xinitrc` normally starts your desktop.

### **CTRL-ALT-'-' and CTRL-ALT-'+'**

Switches to the resolution you want. Check your `/etc/X11/XF86Config` if you don't have any other resolutions available.

### **CTRL-ALT-F1**

Switches back to the text console.

### **xterm -display localhost:0.0 &**

All X applications accept a `-display` command line argument to tell it which X server to display on. Every X program is an X server client. 0.0 represents the screen on F7, 1.0 on F8, etc. localhost is your computer. Its possible and very common to execute an X program on one computer but have it display on some other workstation. localhost could be an IP address or the dns name of any computer running X that is accepting connections from where you run the X client application program.

### **ALT-F7**

Switches back to X 0.0. Using xterm, start XQF or whatever program you use to launch Quake.

### **glqwcl.glx -width w -height h**

Tell your quake launcher to start it like this, where w and h are the resolution you are at. A good resolution is 1024 by 768.

Light effects.

At the console, you can try turning on and off the flashblend function and see which lighting effects you prefer:

```
gl_flashblend 0  
gl_flashblend 1
```

When it is on (1), you may notice colored light effects around flags and special items. In this mode, you may NOT see flashes from weapons and light on the walls from rockets. The exact effects of toggling flashblend may depend on your specific OpenGL installation.

When it is off (0), there may not be any COLORED light, but otherwise all light effects work. Flags and special items have a glow around them and light up surrounding surfaces. Weapons flash when fired and rockets light up everything around them as they travel.

*HAVE FUN!*