The Official TORCS FAQ (frequently asked questions)

0. About this Document.

1. About TORCS in General.

2. Platforms and Requirements.

3. Installation.

4. Problems and Solutions.

5. Problem reporting.

6. Researchers FAQ.

7. Links.

0. About this Document.

0.1 License and copyright.

0.2 Document version.

2012-10-19.

1. About TORCS in General.

1.1 What is TORCS?

TORCS, **The Open Racing Car Simulator**, is a car racing simulation, which allows you to drive in races against opponents simulated by the computer. You can also develop your own computer-controlled driver (also called a robot) in C or C++. TORCS is "Open Source" (GNU General Public License Version 2 or later).

1.2 Who develops TORCS?

The TORCS project has been created by Eric EspiÃ© and Christophe Guionneau. Currently Bernhard Wymann, Christos Dimitrakakis and other contributors continue to develop the project. Visit the official TORCS site for more information.

1.3 Is documentation available?

Yes, there is quite a lot of documentation available. Look at the official TORCS site and on Bernhard's homepage. You can find installation instructions, the robot tutorial, instructions how to use TORCS and more.

1.4 Which features has TORCS?

There are many different cars and tracks, and around 100 opponents to race against. You can steer with a joystick, gamepad or steering wheel, if it is supported by your platform. It is also possible to drive with the mouse or the keyboard. Graphic features lighting, smoke, skid marks and glowing brake disks. The simulation features a simple damage model, collisions, tire and wheel properties (springs, dampers, stiffness, ...), aerodynamics (ground effect, spoilers, ...) and much more. The gameplay allows different types of races from the simple practice session up to the championship. Enjoy racing against your friends in the split screen mode with up to four human players.

1.5 What are the future plans?

First, take the following features not as promise, it can take a very long time till the features are working if at all. Beside the maintenance and bugfixing the following features will probably become implemented: Improved physics, better inside view, replay and networking.

1.6 Why do you not...?

Don't ever forget that we develop TORCS in our spare time and you get it for free. Your wishes, ideas and feature requests are welcome in the torcs-users mailing list. If nobody picks up your idea feel free to contact the project leader and to implement it yourself.

1.7 What do you mean with robot?

In TORCS a "robot" is a piece of program code that drives a car, so all the opponents you choose are robots.

1.8 Why do you not replace GLUT with SDL?

We will switch soon because of portable threading and networking, but for fun I leave the former answer.

Ah, simple question, simple answer. Like you can easily see G-L-U-T has four letters and S-D-L has just three letters, that should be enough proof of superiority, is it? Ok, then look at the version numbers, GLUT has a major 3.7, SDL has a childish 1.2.7, so GLUT is 3.7/1.27 => 2.913385826 times better at least. By the way, I am sure there are numerous technical reasons why not to use SDL for games, got it, bro?

Ok, now back to serious mode. First let us have a closer look at the question, to understand it you need to have an idea what SDL, GLUT and plib provide. In a nutshell, SDL provides framebuffer access, OpenGL context, a thread API, support for input devices and audio output. There are loads of additional services available through additional libraries. GLUT provides an OpenGL context, keyboard and mouse input and simple window management functionality. Now you can see that SDL provides much more functionality than GLUT, but where does e. g. the sound in TORCS come from? Yes, it comes form plib. Plib provides sound, a scene graph, a geometry library, joystick input and much more.

Now you can see what we are talking about. We need plib because of its scene graph, so the only thing we need and SDL could provide is the OpenGL context, input and simple window management. Hmm, sounds familiar, does it? Yes, GLUT fits perfectly, no SDL functionality is needed at the moment. That raises the question why should we replace working code with new code if there is no advantage? Yes, it is simply a crazy idea. Keep in mind, all the libraries are maintained and have an appropriate license. Both FreeGLUT and SDL are not standard, so there is also no advantage.

Conclusion: SDL is a great library but we don't need it, it simply does not fit the requirements. That might change in the future, one never knows.

1.9 Are there official championships?

Yes, you can find offline robot championships here.

1.10 Should I start my own racing sim project?

Definitively not, no! Why? Do a search on sf.net and Google, you will find between 15 and 40 open source car/driving/racing sim projects, and most of them have been "promising" and died. Instead of starting another dead project contribute to the few existing successful ones.

Believe it or not, but your project will very likely be a dead one as well. The reason is quite simple, people underestimate the required effort and the change of their lives during the project lifespan, think about it. Just that you get an idea, TORCS turns now 15...

2. Platforms and Requirements.

2.1 On which operating systems runs TORCS?

Linux (x86, AMD64 and ppc are known to work), FreeBSD, OpenSolaris and Windows 2000/XP/Vista/7.

2.2 What are the hardware requirements?

2.2.1 Robot development.

Minimum: 400MHz CPU, 128MB RAM, OpenGL 1.2 compatible graphics card with 16 MB RAM.
Recommended: 600MHz CPU, 256MB RAM, OpenGL 1.3 compatible graphics card with 64 MB RAM.

2.2.2 Driving yourself.

Minimum: 550MHz CPU, 128MB RAM, OpenGL 1.3 compatible graphics card with 32 MB RAM.

Recommended: 800MHz CPU, 256MB RAM, OpenGL 1.3 compatible graphics card with 64 MB RAM.

2.3 What are the software requirements?

You need recent, working and properly configured OpenGL/DRI drivers. You need also several libraries. To avoid problems you should prefer FreeGLUT over GLUT. Make sure that you use exactly plib version 1.8.5. For AMD64 you need to set the "-fPIC" compiler switch when compiling plib (export CFLAGS="-fPIC", export CPPFLAGS="-fPIC", export CXXFLAGS="-fPIC"). You need as well OpenAL.

3. Installation.

3.1 How do I install TORCS?

You can find installation instructions on the TORCS site and on Bernhard's homepage.

4. Problems and Solutions.

4.1 General problem solving approach.

You may encounter this kind of problems: System configuration problems, build problems and runtime problems. If you face a problem follow these steps:

- Have a look into this FAQ, perhaps your problem is already known.
- Clean up your old TORCS installation, including old environment settings.
- Read the installation instructions carefully, double check your steps.
- Make sure that you have not installed multiple GLU libraries.
- Make sure that you have installed the latest OpenGL/DRI and sound drivers.
- Try to start up TORCS with the "-s" switch to disable multitexturing.
- Make sure that the version number of your ALSA drivers, libraries and utilities match.
- Test your OpenGL setup with tuxracer and bzflag.
- Have a look into the mailing list, perhaps there is already a solution.
- If you get stuck learn how to report a problem and contact the mailing list.

4.2 Build Problems.

4.2.1 ./configure reports "configure: error: Can't find GL/glut.h".

Download and install FreeGLUT or install GLUT which is probably included in your distribution (you need also the header files, which may be in a separate package like glut-devel). I suggest using FreeGLUT.

4.2.2 ./configure reports "configure: error: Can't find plib/ssg.h" or "plib/ul.h".

TORCS requires plib 1.8.5 and its header files. Download and install plib.

4.2.3 TORCS does not compile on Windows.

To compile TORCS in Windows you need VisualStudio 6.0 or 2008 (for the other versions you have to adopt the project files by yourself, but this is usually pretty easy and done quickly). Follow exactly the instructions in the README file to build a release.

4.2.4 FreeGLUT compilation fails with NVidia OpenGL driver.

If you have an NVIDIA driver and FreeGLUT breaks the compilation with complaints about libGL.la, then update your drivers to 5336 (previous libGL.la was broken).

4.2.5 TORCS does not compile with make: 'Make-config' is up to date.

You have some environment variables set like $TORCS_BASE and $MAKE_DEFAULT (to follow the robot tutorial), they are pointing now to the wrong directory. Unset or update the variables suitable for the new directories.

4.2.6 Compiling stops with "error: `GL_TEXTURE0_ARB' undeclared".

If the compiler stops with the above message you have probably OpenGL header files installed which do not define the

GL_TEXTURE0_ARB and other symbols. If you have updated your NVidia drivers to 6xxx you will experience this error (NVidia removed those definitions...). To fix it for TORCS you need to install a gl.h which defines the missing symbols. You can get it from your old (5xxx) NVidia drivers or from Mesa (the problem is not the driver itself).

4.2.7 Compiling fails with "The syntax of the command is incorrect. Error executing c:\winnt\system32\cmd.exe".

You have probably installed your sources in a path which contains blanks or other special characters, e.g. on the desktop. Move your torcs-1.3.6 folder to a location with a "clean" path, a drive is the best bet (e.g. D:\torcs-1.3.6).

4.2.8 ./configure fails in "checking for sin in -lm" on Gentoo.

This is a Gentoo problem, have a look at this report.

4.3 Startup and Runtime Problems.

In several situations TORCS will crash at startup. The cause are often not bugs of TORCS, but a broken installation or buggy OpenGL/DRI drivers.

4.3.1 Broken OpenGL/DRI drivers or '~' file.

```
/usr/local/bin/torcs: line 54: 31895 Segmentation fault
$LIBDIR/torcs-bin -l $LOCAL_CONF -L $LIBDIR -D $DATADIR $*
```

There are two known possibilities which cause such a crash:

- You have buggy OpenGL/DRI driver, you should update to a recent version. Start TORCS with the "-s" option to disable multitexturing.
- You have edited a TORCS XML file (e. g. endrace.xml) and your editor left a copy with an appended "~" (e. g. endrace.xml~). Remove those "~" files.

4.3.2 Multiple GLU libraries.

```
linuxModLoad: ...  modules/graphic/ssggraph.so:
undefined symbol: gluBuild2DMipmaps
/usr/local/bin/torcs: line 50: 21215 Segmentation fault
./torcs -l $LOCAL_CONF $*
```

You have probably two GLU libraries installed on your system and TORCS has been linked to the wrong one, which does not export the "gluBuild2DMipmaps" symbol. First find all your GLU libraries on the system (cd /usr; find . -name "*GLU*"). Then check which one exports the "gluBuild2DMipmaps" symbol (e. g. nm ./lib/libGLU.so | grep gluBuild2DMipmaps). Finally move or remove the GLU libraries which do not define the symbol, such that the linker can't see it. Now recomplie and reinstall TORCS.

4.3.3 Not removed old TORCS version.

```
/usr/lib/torcs/torcs-bin: relocation
error: /usr/lib/torcs/lib/libtgfclient.so: undefined symbol:
_Z15GfParmWriteFilePKcPvPc
```

If you see such a message you have probably not cleaned up your old TORCS installation. Make sure that you have removed all old TORCS files, then recompile and reinstall TORCS.

4.3.4 TORCS reports missing files and crashes.

```
data/fonts/b5.glf: No such file or directory
data/fonts/b5.glf: No such file or directory
data/fonts/b5.glf: No such file or directory
data/fonts/b5.glf: No such file or directory
data/fonts/b7.glf: No such file or directory
data/fonts/b7.glf: No such file or directory
data/fonts/b7.glf: No such file or directory
data/fonts/b7.glf: No such file or directory
data/fonts/digital.glf: No such file or directory
Can't open file data/img/splash-main.png
Can't open file data/img/splash-single-player.png
Can't open file data/img/splash-qrdrv.png
/usr/local/bin/torcs: line 54:6839 Floating point
exception$LIBDIR/torcs-bin -l $LOCAL_CONF -L $LIBDIR -D $DATADIR $*
```

If you see this you have not installed the mandatory data packages or installed it in the wrong location. Please read the installation instructions carefully and reinstall TORCS.

4.3.5 My screen resolution is not supported.

You have a screen resolution which is not available in the selection menu of TORCS (e. g. 1152x864). Exit TORCS and edit the file .torcs/config/screen.xml, change the x, y, window width and height to your values.

4.3.6 I have a problem with the fullscreen mode.

If you switched TORCS to fullscreen you get a scrolling screen or the screen size is not fully utilized. If you use the original GLUT library you should choose the resolution of your screen in the display settings. To fix the problem you can also download and install FreeGLUT, then recompile and reinstall plib and TORCS.

4.3.7 TORCS graphics is slow and it crashes on "READY!"

Select in the TORCS display menu a color depth of 24 or 32 bpp, even if your display runs at 16 bpp.

4.3.8 TORCS on Windows hangs after "READY!"

Shutdown TORCS and restart it, go to the sound menu and disable the sound or try the other backend. The problem is likely caused by some library we use, but it never happened to us, so we could not fix it.

4.3.9 TORCS does not start and reports slDSP:.

Make sure that the sound device is available for TORCS and you use the latest drivers. Perhaps you have a sound daemon running in the background and you need to start TORCS with a wrapper (e. g. artsdsp of KDE) or to disable the daemon.

4.3.10 The cars are flickering.

Probably your depth buffer has just 16 bit, you should have seen a hint at TORCS startup. Try to run your X-server at 24 or 32 bpp, or update/replace your OpenGL/DRI drivers. In Windows set the quality in the OpenGL properties tab to maximum, such that TORCS has a chance to get a 24 bit depth buffer.

4.3.11 I use ALSA and TORCS does not start up.

You get an error message similar like this:

```
slDSP: getBufferInfo: Broken pipe.
```

Although the sound works with the other applications you have messed up your ALSA installation. Make sure that the versions of your ALSA driver, libraries, tools and utilities are matching. This can especially happen if you run a selfcompiled kernel.

4.3.12 TORCS does not start up and complains about display requirements.

```
Visual Properties Report
------------------------
The minimum display requirements are not fulfilled.
We need a double buffered RGBA visual with a 16 bit depth buffer at least.
GLUT: Fatal Error in torcs-bin: visual with necessary capabilities not found.
```

If you use the open Mesa/DRI drivers, update your OpenGL/DRI drivers and try FreeGLUT. It seems to be a bug in the visual matching code of GLUT or OpenGL/DRI. The output of 'glxinfo | grep "OpenGL version"' should report at least "1.2 Mesa 5.0.1".

4.3.13 The racetrack mini map is just a blue square.

You have probably more than one GLU library installed on your system and TORCS is linked with a version that does not match the version defined in the header files. Remove all old GLU libraries, make sure that the header files are matching with the library. Make also sure that you have recent OpenGL headers installed which define the GL_UNSIGNED_INT_8_8_8_8 symbol, then recompile TORCS.

4.3.14 TORCS runs but without textures.

This is probably an OpenGL setup problem, check your installation with bzflag and tuxcart. You have to reinstall the XFree86 libraries, FreeGLUT and your OpenGL drivers.

4.3.15 TORCS on Windows crashes after "new race".

The instruction at 0x0043dda5 referenced memory at 0x00000004. The memory could not be written.

TORCS on Windows crashes with the above message after you have choosen "new race". Probably you have not downloaded all required files or installed them in the wrong place. Please check your TORCS installation carefully.

4.4 Other Problems.

4.4.1 How do I capture a movie?

First you have to configure the video capturing in config/raceengine.xml. The captured frames are stored in the output directory, so you need to have write permission and plenty of space (~ 1GB for one minute, depends of framerate and resolution).

During the simulation you can hit "c" to start and stop the capturing. The frames are stored in the output directory and named this way: torcs-ssss-ffffffff.png, ssss is the sequence number and ffffffff the frame number in the sequence.

Now you need a tool to compose the frames to a movie. If you use mencoder (of the mplayer 0.92) you have first to convert the images to jpg's with the png2jpg tool from src/misc/png2jpg. Now encode the video with mencoder -ovc lavc -lavcopts vcodec=mpeg4:vhq:vbitrate=1800 *.jpg -mf on:fps=25.

4.4.2 Can I develop a robot using Windows?

Yes, you can. Here is a rough guide:

- Download the TORCS source and manage to compile and run it.
- Download e. g. the bt robot package, it is a little enhanced version of the robot developed in the tutorial, so there is good documentation available to understand it.
- Copy the sources into a new directory, e. g. when you choose the name "myrobot" create a directory /src/drivers/myrobot.
- Copy all the bt files into the myrobot directory.
- Now rename all "bt*" files into "myrobot*".
- Edit myrobot.def and also change all bt strings into myrobot.
- The same for myrobot.dsp, .xml, perhaps more, you get the idea.
- Change the "bt" module entry point in myrobot.cpp to "myrobot", also make the strings for the robots names and description match with those in myrobot.xml.
- Add the myrobot project to TORCS, open the project with vc++ and try to compile it.
- If you want to deploy the files you have to do that manually, create a directory in "runtime" where the other robots are and copy the required files there (myrobot.dll, .xml, setup subdirectories etc.).
- Now TORCS should pick up your robot.

4.4.3 Is my joystick/steering wheel/gamepad supported?

If your operating system supports your device it will also work within TORCS. To learn how to setup your input device visit the TORCS site and read the "how to drive" section.

4.4.4 How do I set up a multiplayer game?

TORCS supports split screen multiplayer games with up to four players. Setup:

- Set up the players in the "Configure Players" screen.
- If more than one player uses the keyboard, read this.
- Now choose a race, select the players in the configuration screen.
- Start the race, stop it with "p" to pause.
- Now create the split screens with "]", or remove them with "[".
- If you want to change the view of a "mini screen" point with the mouse on it and set it up as usual (F1-F11, ">", "<", ...).
- When you have set up all views press "p" to end the pause and start the race.
- Have fun;-)

4.4.5 I cannot adjust the joystick/wheel centering.

Before you start TORCS make sure that the joystick/wheel is correctly set up/calibrated for the system. You can use the "jscal" and "jstest" utilities for that, you should find them in your Linux distribution. If this works fine center your axis (e.g. if you have noncentering axis like sliders), the start up TORCS and set up your players and controls. It is important to center the axis before you go to the calibration.

5. Problem reporting.

5.1 When do I report a problem?

You should report a problem after you have carefully read the available documentation and did not find a solution for the problem.

5.2 How and where do I report a problem?

First subscribe to the Torcs-users mailing list (otherwise your messages need to be approved, so it can take long till they become distributed). Send us as much information about the problem as you know. At least provide the following:

- A good description of the problem.
- The TORCS version, did you compile from source?
- The console output of TORCS.
- Your operating system (uname -a).
- The output of "glxinfo -l" (Linux, FreeBSD).
- The /var/log/XFree86.0.log file (Linux, FreeBSD).

Please do not file a bug in the bug reporting section on sourceforge, it is not a support area for users (if you know for sure that you have found a bug in TORCS, then it is ok to report it in the bugtracking system).

6. Researchers FAQ.

6.1 How do I explore the source code?

Use a powerful IDE or IDE/editor plugins for navigating the source code (e.g. ctags, virtual assist x, etc.), and set breakpoints and explore the call stack. A good development environment can speed you up a lot, so invest some time initially for being much faster afterwards.

6.2 Where is the physical simulation code located in the source tree?

Navigate to src/modules/simu, then either simuv2 or simuv3 (simuv2 is the dafault simulation core for TORCS, simuv3 is experimental and not tested for the releases, but might be interesting for you).

6.3 How does the simulation work?

It integrates differential equations with Euler steps, timestep is 0.002 seconds (500Hz, simulation time), look at src/interfaces/raceman.h (#define RCM_MAX_DT_SIMU). The raceengine runs as well with this rate, but the robots are just called all 0.02 seconds (50Hz).

6.4 Can I record/send telemetry data somewhere?

There is no such (maintained) functionality in TORCS currently, but you can easily add it. If you are interested in simulation data you can add some code in the raceengine or in the simulation code, if you are interested in the actions of your robot you can add code there.

6.5 Where is the simulation timestep performed?

The simulation timestep is performed in src/libs/raceengineclient/raceengine.cpp, ReOneStep.

6.6 How often is the robot code called?

The robots are called all 0.02s (50Hz), do not confuse this with the physics/raceengine simulation timestep which is 0.002s (500Hz).

6.7 Is there a command line mode for AI training?

Yes, since version 1.3.3. This mode is useful for testing as well. All you need is a race configuration xml, the easiest way to create one is using TORCS. As example navigate in TORCS to the race configuration, select "Challenge Race", set up a grid without human players. Then exit TORCS and run from the command line (Linux and other Posix like systems):

```
~/torcs_bin/bin/torcs -r ~/.torcs/config/raceman/dtmrace.xml
```

Or if you use Windows:

```
wtorcs.exe -r config/raceman/dtmrace.xml
```

Beware, if you use Windows Vista or higher without a local (->in your users directory) TORCS installation the configuration and result files are placed in the virtual store (e.g. under my Windows 7 64 bit it is below "berni/AppData/Local/VirtualStore/Program Files (x86)/torcs").

Any valid race configuration file can be consumed, so you can either set it up by TORCS, edit it manually or generate it somehow. You can think of these XML files as command files for a state engine, have a look at src/libs/raceengineclient/racestate.cpp.

6.8 How do I extend TORCS such that the AI can restart the race?

The solution presented here is based on a discussion started by Alessandro Ghidotti Piovan on the torcs-devel mailing list. To feed back commands to the race state engine (src/libs/raceengineclient/racestate.cpp) add a field in the tCarCtrl struct to allow the robot to command the race state engine:

```
typedef struct {
    ....
    ....
    int        askRestart;    /** boolean; 1 = robot asked to restart the race */
} tCarCtrl;
```

Modify e.g. raceengine.cpp->ReOneStep() to take action on the command:

```
bool restartRequested = false;

for (i = 0; i < s->_ncars; i++) {
    if(s->cars[i]->ctrl.askRestart) {
        restartRequested = true;
            s->cars[i]->ctrl.askRestart = false;
        }
    }

    if(restartRequested){
        ReRaceCleanup();
        ReInfo->_reState = RE_STATE_PRE_RACE;
        GfuiScreenActivate(ReInfo->_reGameScreen);
    }
```

The robot can now set askRestart when desired, e.g.:

```
if(!car->ctrl.askRestart){
       if (isStuck(car))
        {
            // as your robot tutorial and/or set car->askRestart = true if you need.
        }
    }
```

6.9 How do I cite TORCS?

If you use TORCS for your research, please cite it as follows:
B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, A. Sumner. TORCS: The Open Racing Car Simulator, v1.3.6, 2014.

```
@misc{TORCS,
author = "Bernhard Wymann, Eric Espi{\'e},
Christophe Guionneau, Christos Dimitrakakis, R{\'e}mi
Coulom, Andrew Sumner",
howpublished = "http://www.torcs.org",
title = "{TORCS}, {T}he {O}pen {R}acing {C}ar
```

7. Links.

7.1 TORCS.

- The main TORCS site www.torcs.org.
- The TORCS-users mailing list.
- TORCS official championships www.berniw.org/trb.
- Bernhard's site www.berniw.org.

7.2 Libraries and Drivers.

- FreeGLUT freeglut.sourceforge.net.
- Plib plib.sourceforge.net.
- OpenAL www.openal.org.

Up