

CHAI 3D: setup and introduction

G. Aiello, M. Ahmadi, E. Zoller, M. Eugster, I. Susic, G. Kosa, and G. Rauter*

Bio Inspired RObotics in Medicine Lab, University of Basel, Basel

November 2016

E-mail: gregorio.aiello@unibas.ch

Abstract

In this small tutorial it is provided a step by step guide to install and use the famous haptic libraries CHAI3D, first a description will be provided, then the setup and example code are presented.

Installation

In the following sections we provide a brief description of the Installation procedure of the most important files and directories that compose CHAI3D.

First step is to download the packages from:

Download CHAI3D

Then unzip the folder and open the solution file relative to your Visual Studio version, build it and check if it returns errors, if everything goes fine you are ready to use CHAI 3D.

Main Directories

Here are presented the main directories of CHAI3D, after a brief introduction we will dive in and see what is inside:

- `./src`: This directory contains the entire source code of the CHAI3D core framework. We will come back to that one in the Source Code section where we will explore the organization of the different source files.
- `./external`: This directory contains source code to third-party libraries that are used by CHAI3D. Some of these libraries can be enabled or disabled by selecting the desired options in file `CGlobals.h`
- `./modules`: This directory contains all third party extension modules for CHAI3D. The default modules that come with CHAI3D are the ODE wrapper for the open source dynamics engines ODE, the GEL framework for modelling deformable bodies and the V-REP plugin for the Coppelia Robotics V-REP simulator. Each module shares a similar structure that include the source files, project files, examples, and documentation.
- `./lib`: This directory contains the CHAI3D library files that result from compiling the CHAI3D source code.
- `./examples`: This directory contains the project files and source code for the various examples provided with CHAI3D.
- `./bin`: This directory contains the executable files that result from compiling CHAI3D with all of its examples. You may also find in that directory dynamic library files (`.dll` or `.dylib`) which support third-party haptic devices. A resources directory also includes all the media files that used by the different examples.
- `./doc`: This directory contains all documentation related to CHAI3D. The documentation is generated using Doxygen and is provided inlined in every source file of the framework.
- `./templates`: This directory contains three different project templates that use GLUT,

Qt, and the Microsoft .NET frameworks. These templates are very useful if you want to create a new project from scratch.

- `./extras`: This directory contains additional third-party libraries that are used by the examples provided with CHAI3D. They are not required to build the CHAI3D core library, but are necessary for the CHAI3D example executables to build and run properly.
- `./utils`: This directory contains utilities that convert font and images to source code that can be included and built into CHAI3D.

Source Folder

The source code is located in the `/src` directory. It contains the main `chai3d.h` header file and a set of subdirectories with all the source files organized by topic.

- `./src/audio`: This directory contains all the files that provide sound support to CHAI3D. Details are available in the audio section of the documentation.
- `./src/collisions`: This directory contains the files that handle collision detection between rays and geometrical primitives such as points, lines, and triangles. Details are available in the collisions section of the documentation.
- `./src/devices`: This directory contains support for all supported haptic devices. Details are available in the devices section of the documentation.
- `./src/display`: This directory contains files related to framebuffers and virtual cameras which are used to render scenes of the world in viewports or window displays. Details are available in the display section of the documentation.
- `./src/effects`: This directory contains all the classes that model haptic effects (e.g friction, vibration, surfaces, etc...) Details are available in the effects section of the

documentation.

- `./src/files`: This directory contains support for loading and saving 3D models, 2D images, 2D videos, and audio files. Details are available in the files section of the documentation.
- `./src/forces`: This directory contains classes that implement force rendering algorithms such as potential fields and finger-proxy based methods. Details are available in the forces section of the documentation.
- `./src/graphics`: This directory contains all files related to graphic support. These include support for videos, fog, fonts, images, image arrays, videos, etc... Data structures for handling arrays of points, segments, and triangles are also included. Details are available in the graphics section of the documentation.
- `./src/lighting`: This directory contains all classes for modelling light sources and shadows. Details are available in the lighting section of the documentation.
- `./src/materials`: This directory contains all classes for modelling materials and textures. Details are available in the materials section of the documentation.
- `./src/math`: This directory contains the basic code for handling 3D vectors, matrices and quaternions. You will also find a number of functions for performing simple geometrical computations. Details are available in the math section of the documentation.
- `./src/resources`: This directory contains a collection of files for precompiled fonts and shaders that are used for graphic purposes. Details are available in the resources section of the documentation.
- `./src/shaders`: This directory contains all classes for supporting vertex and fragment shaders. Shaders are used for implementing advanced graphic rendering effects. Details are available in the shaders section of the documentation.

- `./src/system`: This directory contains several useful classes for handling strings, mutexes, and threads. Details are available in the system section of the documentation.
- `./src/timers`: This directory contains classes that model a precision clock and frequency counters for accurate time measurement. Details are available in the timers section of the documentation.
- `./src/tools`: This directory contains classes that implement haptic tools which connect haptic devices to avatars that interact with objects in the virtual environment. Details are available in the tools section of the documentation.
- `./src/widgets`: This directory contains different classes for rendering 2D widgets such as text labels, dials, bitmap images, scopes, etc... Details are available in the widgets section of the documentation.
- `./src/world`: This directory contains the classes that are used to compose and describe a 3D scene. These include the world itself, shape primitives, meshes, volumes, and other 3D entities which can be rendered both graphically and haptically. Details are available in the world section of the documentation.