Moai Qt Host
User manual

February 7, 2013

Contents

Chapter 1

Introduction

1.1 Moai Qt Host

Moai Qt Host is a host application for Zipline Games' open source Moai game development framework (`http://getmoai.com/`) for desktop computers.

Using Moai to run your Moai SDK/Lua game scripts is easy with whichever way you have installed Moai Qt Host: Just give your Lua script as argument(s) and run.

1.2 Licensing

Moai Qt Host is open source, but is implemented with open source version of Qt, so Qt licensing rules (`http://qt.digia.com/Product/Licensing/`) apply when releasing games on Moai Qt Host. The license of Moai SDK applies as well.

Chapter 2

Setting up Moai Qt Host

To set up Moai Qt Host you have two options. You can

(a) Dowload Moai Qt Host binary and use it as is, or

(b) Compile Moai SDK and the Moai Qt Host yourself.

2.1 Option A: Download binary

This is the simplest way to go. Download Moai Qt Host binary. You must have the necessary Qt libraries on your computer (see 3.1). The binaries are located at *"moaiqthost/bin/"*-folder.

2.2 Option B: Compile Moai Qt Host yourself

This section describes how to do the debug build of Moai Qt Host. The release build is built similarly. Just make sure you build both the Moai SDK and the Moai Qt Host with same settings.

1. Set up Qt: the build has been tested with the open source version of Qt SDK 1.2.1 (libraries version 4.8.1, Qt Creator version 2.4.1), downloadable from `http://qt-project.org/`. On Windows and OS X you must use the 32-bit version of Qt libraries (on OS X you need to build these yourself). The Linux build has only been tested with 64-bit libraries.

2. Download Moai Qt Host source.

3. Download the Moai SDK. The supported version is 1.3 (Build 160). The source can be found from:

   (a) **On Windows and OS X**: `https://github.com/moai/moai-dev`
   (b) **On Linux**: `https://github.com/spacepluk/moai-dev` (the supported version is dated at January 22nd 2013)

4. Place the Moai SDK source to *"moaiqthost/src/moaipackage"* folder.

5. Compile Moai SDK. Note that Moai SDK and the host must have the same compile configuration (for more information, refer to section 3.2).

   (a) **On Windows**: build with Microsoft Visual Studio 2010, according to instructions in *"moaiqthost/src/moaipackage/vs2010/README.txt"*.
   (b) **On OS X**: build with Xcode or Clang, according to instructions in
   "'moaiqthost/src/moaipackage/xcode/README".
   (c) **On Linux**: build according to instructions in
   "moaiqthost/src/moaipackage/README.md".

6. Copy the compiled Moai SDK libraries to *"moaiqthost/src/moaiqthost/libs"* -folder. You can find the libraries from:

   (a) **On Windows**: *"moaiqthost/src/moaipackage/vs2010/bin/Win32/Debug"*
   (b) **On OS X**: *"/Users/<username>/Library/Developer/Xcode/DerivedData /libmoai-<random text>/Build/Products/Debug"*
   (c) **On Linux**: inside the subfolders of *"moaiqthost/src/moaipackage/build/src/"* and *"moaiqthost/src/moaipackage/build/3rdparty/"* – folders.

7. Open the Moai Qt Host –project from *"moaiqthost/src/moaiqthost/moaiqthost.pro"*.

8. Configure the Moai Qt Host build if needed (refer to chapter 3).

9. In Qt Creator under "Projects", make sure you are using the same compiler that you built Moai SDK with:

   (a) **On Windows**: MSVC2010
   (b) **On OS X**: GCC 4.2
   (c) **On Linux**: GCC 4.6.3

10. Build. The resulting executable file is named "moaiqt", and will appear in:

   (a) **On Windows and OS X**: *"moaiqthost/src/moaiqthost/Debug"* – folder
   (b) **On Linux**: *"moaiqthost/src/moaiqthost/"* – folder

2.3 About audio

Moai offers two options for implementing audio. Zipline recommends using the open source UNTZ audio library.

The second option is to use FMOD audio library, which is not free for commercial development. Enabling either option in Moai Qt Host is described in chapter 3. FMOD Ex is the current FMOD version.

Note: Fmod requires runtime libraries. These are fmodex.dll on Windows and libfmodex.dylib on OS X.

Chapter 3

Using and configuring Moai Qt Host

3.1 General

In general, just give your Lua scripts for the host as command line arguments.

Moai Qt Host requires Qt libraries in order to work, so you must make sure that they are available. To do this you can, for example, include the libraries in the same folder as the Moai Qt Host executable, or add the libraries path to environment variables.

3.2 Compile Configurations

Moai Qt Host can be configured for different purposes by qmake configurations. Different parameters are explained in table 3.1 on page 8. The ”Dependency libraries”-column refers to the libraries that should be copied to ”libs”-folder (refer to section 2.2 step 6) when using corresponding configuration.

Enabling each parameter is done by adding them to the `CONFIG`-variable in *moai_root.pro*-project file. For Windows default configuration, which has the `luaext` and `particle_presets` –parameters enabled, this would look like following:

```
CONFIG += luaext particle_presets
```

Default configuration for OS X and Linux are:

OS X: `CONFIG += luaext particle_presets untz`
Linux: `CONFIG += untz`

To get the required libraries and to ensure compatibility, the Moai SDK should be built with same kind of configuration as the Moai Qt Host. So, for example, if you want to use FMOD Ex –sound system, you should build the Moai SDK with FMOD Ex configuration and use the libraries from that build, in addition to setting the `"fmod_ex"`-parameter for Moai Qt Host build.

Table 3.1: Configuration parameters for Moai Qt Host

Parameter	Description	Dependency libraries
audiosampler	Enables the MOAIAudioSampler-extension. Not fully functional at the time of writing this document.	—
debugger	Enable the Moai SDK harness for external debuggers. Using 3rd party debuggers (such as Zero-Brane Studio) with Moai Qt Host has not been tested and might not work.	`moaiext-debugger`
fmod_designer	Use FMOD Designer as the sound system (Not tested).	From Moai SDK: `moaiext-fmod-designer` From FMOD: `fmodex_vc`
fmod_ex	Use FMOD Ex as the sound system (Not tested on Linux).	**Windows & Linux**: From Moai SDK: `moaiext-fmod-ex` From FMOD: `fmodex_vc` **OS X**: From Moai SDK: `moai-osx-fmod-ex` From FMOD: `libfmodex`
luaext	Enable the Lua-extensions for Moai SDK.	**Windows & Linux**: `moaiext-luaext` **OS X**: `moai-osx-luaext`
particle_presets	Use custom handler (`ParticlePresets`-C++-files) for particle systems.	—
untz	Use UNTZ as the sound system.	**Windows**: `moaiext-untz` **OS X**: `moai-osx-untz` **Linux**: `moaiext-untz` and `untz`

Chapter 4

Glossary

Term	Explanation
C++	Statically typed, free-form, multi-paradigm, compiled, general-purpose programming language.
FMOD	FMOD is a programming library and toolkit for the creation and playback of interactive audio.
Lua	Lightweight multi-paradigm programming language designed as a scripting language
Microsoft Visual Studio 2010	Integrated development environment from Microsoft.
Moai	Development and deployment platform designed for the creation of mobile games on iOS and Android smartphones. The Moai platform consists of Moai SDK, an open source game engine, and Moai Cloud, a cloud platform as a service for the hosting and deployment of game services. Moai developers use Lua, C++ and OpenGL, to build mobile games that span smartphones and cloud.
MSVC2010	Microsoft Visual C compiler.
qmake	qmake is a utility that automates the generation of Makefiles. Makefiles are used by the program make to build executable programs from source code. qmake is distributed via the Qt toolkit.
SDK	Software Development Kit.
UNTZ	Open source cross-platform audio library.

Term	**Explanation**
Xcode	Xcode is an Integrated Development Environment containing a suite of software development tools developed by Apple for developing software for OS X and iOS.