Install CMake

Asset-Importer-Lib can be build for a lot of different platforms. We are using cmake to generate the build environment for these via cmake. So you have to make sure that you have a working cmake-installation on your system. You can download it at https://cmake.org/ or for linux install it via

```
sudo apt-get install cmake
```

Get the source

Make sure you have a working git-installation. Open a command prompt and clone the Asset-Importer-Lib via:

```
git clone https://github.com/assimp/assimp.git
```

Build instructions for Windows with Visual-Studio

First you have to install Visual-Studio on your windows-system. You can get the Community-Version for free here: https://visualstudio.microsoft.com/de/downloads/ To generate the build environment for your IDE open a command prompt, navigate to your repo and type:

```
> cmake CMakeLists.txt
```

This will generate the project files for the visual studio. All dependencies used to build Asset-IMporter-Lib shall be part of the repo. If you want to use you own zlib.installation this is possible as well. Check the options for it.

Build instructions for Windows with UWP

See https://stackoverflow.com/questions/40803170/cmake-uwp-using-cmake-to-build-universal-windows-app

Build instrcutions for Linux / Unix

Open a terminal and got to your repository. You can generate the makefiles and build the library via:

```
cmake CMakeLists.txt
make -j4
```

The option -j descripes the number of parallel processes for the build. In this case make will try to use 4 cores for the build.

If you want to use a IDE for linux you can try QTCreator for instance.

CMake build options

The cmake-build-environment provides options to configure the build. The following options can be used:

- **BUILD_SHARED_LIBS (default ON)**: Generation of shared libs (dll for windows, so for Linux). Set this to OFF to get a static lib.
- **BUILD_FRAMEWORK (default OFF, MacOnly)**: Build package as Mac OS X Framework bundle

- **ASSIMP_DOUBLE_PRECISION(default OFF)**: All data will be stored as double values.
- **ASSIMP_OPT_BUILD_PACKAGES (default OFF)**: Set to ON to generate CPack configuration files and packaging targets
- **ASSIMP_ANDROID_JNIIOSYSTEM (default OFF)**: Android JNI IOSystem support is active
- **ASSIMP_NO_EXPORT (default OFF)**: Disable Assimp's export functionality
- **ASSIMP_BUILD_ZLIB (default OFF)**: Build your own zlib
- **ASSIMP_BUILD_ASSIMP_TOOLS (default ON)**: If the supplementary tools for Assimp are built in addition to the library.
- **ASSIMP_BUILD_SAMPLES (default OFF)**: If the official samples are built as well (needs Glut).
- **ASSIMP_BUILD_TESTS (default ON)**: If the test suite for Assimp is built in addition to the library.
- **ASSIMP_COVERALLS (default OFF)**: Enable this to measure test coverage.
- **ASSIMP_WERROR(default OFF)**: Treat warnings as errors.
- **ASSIMP_ASAN (default OFF)**: Enable AddressSanitizer.
- **ASSIMP_UBSAN (default OFF)**: Enable Undefined Behavior sanitizer.
- **SYSTEM_IRRXML (default OFF)**: Use system installed Irrlicht/IrrXML library.
- **BUILD_DOCS (default OFF)**: Build documentation using Doxygen.
- **INJECT_DEBUG_POSTFIX(default ON)**: Inject debug postfix in .a/.so lib names
- **IGNORE_GIT_HASH (default OFF)**: Don't call git to get the hash.
- **ASSIMP_INSTALL_PDB (default ON)**: Install MSVC debug files.