Engine186 Linux Port Documentation

Student: Nikole Leopold

Supervisors: Johannes Unterguggenberger

September 22, 2019

1 INTRODUCTION

Engine186: A rendering engine, implemented with C++17 and OpenGL 4 is an open source project created and maintained by Johannes Unterguggenberger. Its source code can be found at [Joh18a].

A Linux port of Engine186 was made by Nikole Leopold. The Linux port source code can be found at [Joh18b].

This document provides documentation on how to use the Linux port and describes the changes made to the original upstream code. Familiarity with Engine186 usage is assumed.

2 DEPENDENCIES

Install the following shared (dynamically linked) libraries via your linux package manager:

- glm
- glfw3
- assimp
- anttweakbar

For the remaining libraries, source code is provided by this git repository. The following thus do **not** need to be installed:

- glad

- stb_image

- filewatcher

3 BUILD INSTRUCTIONS

1. Install dependencies.

2. Via your terminal shell run following commands in the repository root directory:

```
mkdir build-linux  (if directory does not exist)
cd build-linux
cmake ..
make
```

4 CHANGES MADE

The following is a description of all the changes that have been made to the original Windows platform source code of Engine186.

- Fixed UTF-16LE encoding in source files generated by Visual Studio converted to ASCII. Visual Studio in newer versions seems to encode files in UTF-16LE. This is unnecessary for source files which should be ASCII only and entails compatibility issues with other platforms. Make sure to convert source files to US-ASCII or ISO/IEC 8859-1 or even UTF-8 using iconv or just copy the code to new files. For details refer to [Mic17].

- Setup CMakeLists file and CMake modules used to locate libraries

- Fixed `glm/simd/platform.h` to include option for GCC 7.3 (before listed options included versions 7.2 or earlier and version 8, but not 7.3). For details refer to [g-t18].

- Using CMake module cotire to have "e186.h" as a precompiled prefix header (code inserted at beginning of every file), like Visual Studio does via `stdafx.h` which in the original source includes `e186.h`. Without this, a lot of #includes would be missing in all files. For details refer to [cot18a] and [cot18b].

- Fixed some minor bugs reported by GCC during compilation, that are not reported by the Visual Studio compiler. These arise because Visual Studio does not fully comply to the C++ standard.

- CMake `add_definitions(-DGLFW_INCLUDE_NONE)` to add `#define GLFW_INCLUDE_NONE`. This is needed because GLAD provides OpenGL headers, so GLFW should not include them again.

- Made `ExceptionWithCallstack` a subtype of `std::runtime_error` – which is a subtype of `std::exception` – rather than `std::exception` itself. This is necessary because `std::exception` takes no string message argument in the official C++ standard version, MSVC ignores the standard and have their own implementation which takes a string. To be portable, the `std::runtime_error` subtype should be used. For details refer to [Sta18].

- Replaced function calls `sprintf_s`, which is a MSVC-only implementation, with the official standard implementation `snprintf`. Both write formatted output to a C string in a pointed buffer and have the same parameters.

- In `e186.h` added `#ifdef` for `#include <stb_image.h>`:

```
#ifdef __WIN32
// stb_image.h is a single-header lib.
// on linux it is included in Tex2D.cpp
// according to https://github.com/nothings/stb/blob/master/stb_image.h
// on windows a static library version is used
#include <stb_image.h>
#endif
```

  In `src/Tex2D` added:

```
#ifdef __linux__
#define STB_IMAGE_IMPLEMENTATION
#include "stb_image.h"
#endif
```

- Added the GLAD OpenGL loader provided by the original project to the CMake build process. GLAD itself need not be installed, it is provided with the project source and the OpenGL loader is generated by it and is platform independent. For details refer to [Dav19].

REFERENCES

[cot18a]   cotire. cotire: CMake module to speed up builds., 2018.

[cot18b]   cotire. cotire: using a manually maintained prefix header instead of the automatically generated one, 2018.

[Dav19]   Dav1dde. glad: Multi-Language Vulkan/GL/GLES/EGL/GLX/WGL Loader-Generator based on the official specs., 2019.

[g-t18]   g-truc. Include Error, all types "have no member named 'data'", 2018.

[Joh18a]   Johannes Unterguggenberger. Engine186: A rendering engine, implemented with C++17 and OpenGL 4, set-up for Visual Studio 2017, 2018.

[Joh18b]   Johannes Unterguggenberger, Nikole Leopold. Engine186 Linux Port, 2018.

[Mic17]   Microsoft. Visual studio 2017 creates utf-16 source code files, 2017.

[Sta18]   StackOverflow. How to throw std::exceptions with variable messages?, 2018.