Инструкция по взаимодействию с Emscripten OpenGL

Автор:
Касьян Александр Иваночвич

Дата:
October 18, 2020

Оглавление

1 Вводное слово

В данной секции я бы хотел обозначить некоторые моменты, которые могут сбивать.

- Данный, как и все остальные документы я оформляю с помощью LaTex, в связи с этим ни doc ни docx файлы предоставить не могу (если таковые потребуются). Все source-файлы находятся на GitHub'e в соответствующей папке *doc*
- Я заранее прошу прощения, но у меня есть некоторые проблеммы с русским языком, в связи с этим данный и все последующие файлы могут быть (скорее всего будут) написаны не очень грамотно и с большим использованием английских слов.
- Данный документ будет пополняться паралельно с ходом выполнения проекта. И все изменения я буду вносить в секцию «Что изминилось и новый функционал»

2 Что изминилось и новый функционал.

- Я убрал `git submodules`, чтобы избежать ошибок в будущем. В данном проекте используются:
  - GLFW 3.3.2 API для OpenGL.
  - GLEW 2.1.0 для низкоуровневого взаимодействия с OpenGL.
  - SOIL 1.20 библиотека для загрузки изображений.
  - GLM 0.9.9.8 библиотека для удобной работы с линейной алгеброй и OpenGL.
  - EMSDK 2.0.7 emscripten compiler.
  - YAML-CPP 0.6.3 библиотека для работы с YAML файлами.

3 Установка, Сборка и Запуск

Я сразу хочу подметить, что проект не создавался как кроссплатформенное решение, т.к. я считаю, что это глупо использовать для кроссплатформы OpenGL (так считаю не только я: ссылка на видео), т.к. у каждой платформы есть более подходящие спецификации, с более комплексным и гибким API. Поэтому проект работает на платформе Windows (а точнее, компилятор MSVC) и Windows/*nix с компилятором emcc (Emscripten compiler). За поведение при использовании других компиляторов я ответственности не несу.

3.1 Требования перед сборкой

1. Иметь установленный Git

Сборка и запуск с помощью Docker

1. Если вы хотите самый лёгкий и быстрый запуск, тогда лучше просто иметь Docker
2. Если вы по какой-то причине не установили Docker. Установите Docker.

Сборка и запуск без Docker

1. Если вы решили пропустить пункт с Docker, тогда потребуются CMake.
2. Так же нужно установить Emscripten compiler
3. Необходимо иметь Python3
4. Если вы пользователь Windows - лучше иметь WSL или воспользуйтесь nmake

3.2 Установка

Варианта установки будет два, первый - я скину zip, второй - с помощью git. Лучше пользоваться git, т.к. я использую *git submodule*, а это как-то более нативно, что-ли. Здесь я рассмотрю чисто git. Вся работа происходит через cmd/terminal!

1. `> git clone https://github.com/JuiceFV/Emscripten_OpenGL.git`
2. `> cd Emscripten_OpenGL`
3. Сейчас, если вы в cmd наберёте `tree .` то получите более развёрнутую версию этого дерева (здесь приведено всё самое нужное для сборки):

```
Emscripten_OpenGL
├── application
├── cmake
├── config
├── doc
├── CMakeLists.txt
├── Dockerfile
├── docker-compose.yaml
└── requirements.txt
```

4. Далее выбирите нужный вам способ. (Да я всё еще деклссирую не Докер вариант)

Docker

(a) Запустите Докер:

- **Linux:** `sudo service docker start`
- **Windows:** `Win + S` → `"Docker Desktop"` → Ждём пока значёк кита на панели задач не стабилизируется → запускаем cmd.

(b) Настройте файлы конфигурации

(c) Находясь в корневой дериктории проекта. Разверните сервис командой `docker-compose up application`

(d) Как только консоль зависнет на строчке `Runing the server. Follow the link http://<localhost>:<port>`. localhost и port устанавливается мануально в файле конфигурации. По дефолту localhost=localhost и port=8080. Перейдите по этой ссылке.

Not Docker

(a) А может всё таки докер?

(b) (a)????!!!! Пожаааааааалуйста!!!!!

(c) Ладно, тогда небольшое разъяснение. Я не заморачивался на тему запуска emcc через Visual Studio. так что здесь тоже появляется две опции. 1 - Запуск кода через VS без веб ассемблирования. 2 - запуск через emcc с веб ассемблированием.

Компиляция с помощью EMCC. Желательно делать через WSL
Перед сборкой у вас ОБЯЗАТЕЛЬНО должен быть установлен emsdk. А вообще, ещё раз ознакомтесь с pre-requirements

i. Создайте новую папку в корневой дериктории проекта `mkdir build` и перейдите в неё `cd build`

ii. Собираем проект. У emsdk есть возможность собирать с помощью конфигурационных файлов, таких как CMake. Собираем: `emcmake cmake -DFOR_EMSDK=ON ..`

iii. На WSL должен появится Makefile. Его тоже нужно собрать: `emmake make`

iv. Если вы всё же не получили Makefile, т.к. вы не используете WSL, тогда перейдите по ссылке, чтобы собрать и запустить проект без WSL.

v. Все исполняемые файлы сохранаяются в **application/out**

vi. запускаем сервер: `python -m http.server`, переходим по ссылке `http://localhost:8000` и открываем Emscripten_Graphics.html

Компиляция без EMCC

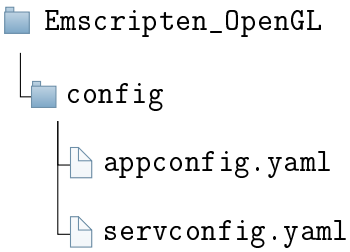
i. Создайте новую папку в корневой дериктории проекта `mkdir build` и перейдите в неё `cd build`

ii. Собираем проект. `cmake ..`

iii. Открываем Emscripten_Graphics.sln и запускаем

3.3 Файл конфигурации

Вся конфигурация приложения происходит через YAML. Как по мне это удобно и не требует прямого взаимодействия с кодом. Файлы конфигурации находятся в папке *config*:



- **appconfig.yaml** - данный файл репрезентует конфигурацию самого приложения (ширина/высота окна, путь до модели и т.д.)
- **servconfig.yaml** - конфигурация сервера на котором будет запускаться приложение (host, port и т.д.).

3.3.1 Конфигурация приложения

Данный файл называется *appconfig.yaml*. Менять название нельзя. Либо в коде нужно напрямую указывать путь к файлу.

```yaml
application:
    window:
        width: 640
        height: 480
```

- `window` - параметры окна
- `window:width` - ширина окна
- `window:height` - высота окна