
ochre Documentation

Release 0.0a0.0

Sunny Zhou (quatridunull) and Jeffrey Souza (JeffreySouza)

July 15, 2016

| | | |
|-----------|----------------------------------|-----------|
| 1 | Installation | 3 |
| 1.1 | Dependencies | 3 |
| 1.2 | Downloading ochre | 3 |
| 2 | Hello ochre! | 5 |
| 2.1 | Library Structure | 5 |
| 2.2 | Implementing Your Code | 5 |
| 3 | Engine class | 7 |
| 4 | Window class | 9 |
| 5 | Sprite class | 11 |
| 6 | ShaderProgram class | 13 |
| 7 | SpriteHelper class | 15 |
| 8 | Contributing | 17 |
| 8.1 | Useful Resources | 17 |
| 9 | Library Structure | 19 |
| 9.1 | Backend | 19 |
| 9.2 | Game | 19 |
| 9.3 | Include | 19 |
| 9.4 | Lib | 19 |
| 9.5 | Src | 19 |
| 10 | Writing Documentation | 21 |
| 10.1 | Code commenting | 21 |
| 10.2 | README.md | 21 |
| 10.3 | Read the Docs | 21 |
| 11 | License | 23 |
| 12 | Indices and tables | 25 |

ochre is a 2D game engine using OpenGL and SDL2, implemented in c++. It is being developed and maintained by NAndDerthal Games.

The documentation is organized into several sections:

- *Introduction*
- *Getting Started*
- *API*
- *Backend*
- *About*

Information for developers is available here as well:

- *Developer Documentation*

Installation

1.1 Dependencies

External dependencies are listed below, and the installation instructions for these can be found on the respective libraries' websites. Note that if your machine's version of OpenGL is insufficient, you can usually solve this by updating your graphics drivers and/or installing [Mesa](#).

- OpenGL 3.3 or higher
- SDL2
- [GLEW 1.1](#)
- [SOIL](#)

1.2 Downloading ochre

To download the source, go to the location in which you want to save your game and clone the repo using [git](#). (The git repository can be viewed on GitHub [here](#).)

```
git clone https://github.com/NANDerthal/ochre.git
```

The ochre engine does not have a GUI and is intended to be used in its source form, so no further installation is required.

To test that your environment is configured correctly with the dependencies listed above, go to the ochre folder and make the test as follows:

```
cd ochre/  
make  
./main.out -t
```

If all went well, you should see a window with nothing in it. It will close itself after two seconds.

Hello ochre!

2.1 Library Structure

When developing with ochre, you will primarily work in the `game/` folder. This is where all of the code you implement will go. When running `make` from the ochre folder it will compile `game/Game.cpp` and `game/Game.h` in order to run your game. Currently those files will contain our sample game code.

Inside the `game/` folder, you will also find the skeleton code `Game-template.cpp` and `Game-template.h` that outline the essential parts of the `Game` class.

2.2 Implementing Your Code

After successfully completing the test, you can remove the current `Game.cpp` and `Game.h`, or move them somewhere else if you would like to refer to them later. Then rename `Game-template.cpp` to `Game.cpp` and `Game-template.h` to `Game.h`.

Engine class

Window class

Sprite class

ShaderProgram class

SpriteHelper class

Contributing

We appreciate any and all efforts to help improve ochre. However, currently, we are still building the core of the engine, and need to finish this before considering any source contributions. Once this is complete, we will add documentation about how to contribute, so stay tuned!

8.1 Useful Resources

Although it is entirely possible to assist in development without knowing about the underlying libraries used in ochre, it will likely become necessary to learn more about OpenGL and SDL2 if you want to contribute more extensively.

To that end, here are some resources that the current developers have found to be especially useful.

OpenGL:

- [Learn OpenGL](#)

SDL2:

- [Lazy Foo's SDL2 tutorials](#)

Library Structure

Within the `ochre` folder, the library is organized as follows:

- `backend/`: functionality which users of ochre should not need to access. These generally are layers of abstraction between underlying libraries (such as OpenGL and SDL) and the ochre API.
- `game/`: the `Game` class definition and any user-defined functionality, as well as testing functionality in the sample `Game.h` and `Game.cpp` files. In general, actual ochre library functionality should not go here.
- `include/`: header files for functionality which is part of the ochre API, and which the user can and should interact with. All underlying library functions and types should be abstracted away from API-interactable functionality here (return and input types of public members).
- `lib/`: `cpp` files defining the declarations given in `include/`. No new functionality should be defined here.
- `src/`: files which include a `main()` function. Most importantly, contains `main.cpp`, which runs all the code the user defined in `game/`. It should not be necessary to modify `main.cpp`.

9.1 Backend

9.2 Game

9.3 Include

9.4 Lib

9.5 Src

Writing Documentation

When developing for ochre, make sure to write documentation for the functionality you introduce and/or change. There are three main methods of documentation: *code commenting*, *the GitHub readme*, and *Read the Docs* (this website).

Our goals and standards for each method of documentation are outlined below.

10.1 Code commenting

We are currently deciding on a consistent style for code commenting, and will update this section once we have done so.

10.2 README.md

This is the readme on the [ochre GitHub repo](#). It is intended to give a very brief overview of what ochre is, with brief instructions and links for how to get started and get help. It is not intended to be an extensive or detailed guide or reference.

10.3 Read the Docs

The documentation you are reading right now is hosted on [readthedocs.io](#). We have attempted to make this documentation both easily accessible and detailed enough to satisfy the needs of more advanced users who want to delve into the engine source.

License

Indices and tables

- `genindex`
- `modindex`
- `search`