Welcome To The Oats Framework

Guide / Documentation

The Oat’s framework is a very lightweight and small OpenGL based rendering framework.

It is supposed to be very barebones and well very open to ways of usage and implementation.

The code at the moment of typing this is a huge mess but is functional and performs well enough.

Framework Features?

* Custom Framebuffers for post processing
* Easy Shader Loading
* Fairly well abstracted OpenGL classes
* A basic 2D Renderer
* A simple SDL2 based Windowing and Input Managing System
* A Text Renderer
* A built in Timer class that offers precision good enough for most things
* A easy and simple way to query OpenGL context attributes and information
* OpenGL 3.3 and OpenGL 4.5 support

Dependencies

* GLM : OpenGL Mathematics Library by GL-Truc
* SDL2 : Simple Direct Media Library 2
* GLEW : OpenGL Extension Wrangler ( feel free to replace )
* SDL2 Image : Simple image loading
* OpenGL : Open Graphics Library
* Free Type 2 : Font Rendering

Road Map ?

* DirectX Renderer Support (The least of my concerns)
* Particle and Sprite Batcher ( Particles are more important )
* 3D Support ( This is primarily a 2D Framework however )
* Huge code clean up and cleaner API ? Yes.

How To Use

This framework was developed with the intention of making simple 2D games and possibly slightly more advanced ones.

It contains the bare components that most games should require. At the moment of writing the code base and API usage is fairly in

Consistent but it’s not impossible to use and is certainly still viable for actually development.

Generally

All the classes are easy to use and have a few functions that can be called for effectiveness.

Function arguments are very clear and not mystical, and all use RAII effectively for initializing and freeing memory.

Typically the function you would call is fairly obvious from the class it is ( For Example : glTexture. Load From Texture, or

Renderer2D Draw, TextRenderer Render(text) etc ).

The glQuery Functions however are not in a class and are C based with a struct and procedural style. However do not be deterred because

It is still extremely easy to use with a few function calls.

The general idea of this framework is to make the work flow obvious to work with and not impossible to work with.