Scribble2D: 2D-Renderer

Mason Luna
December 4, 2024

1: Intro

1.1: Purpose

- The purpose of the VisionDoc is to provide an overview of Scribble2D as a project. This document covers:
    - Problems and constraints
    - Product overview
    - Product Features

1.2: Scope

- Scribble2D will be a two-dimensional rendering engine that can be compiled into a static library to be used for Windows applications. Scribble2D will eventually support multiple graphics APIs, but initially only OpenGL support will be implemented.

2: Overview

2.1: Perspective

- Scribble2D will be a high-level graphics framework that compiles into a static library to be included in C++ projects as a two-dimensional rendering engine. The primary focus of this rendering engine is to be used in mathematics simulations and 2D games.

2.2: Capabilities

- Using this library will allow the user to create real-time two dimensional scenes that supports moving objects, a moving camera, lighting, and other capabilities often seen in a 2D Renderer

2.3: Assumptions and Dependencies

- Scribble2D should be built with the ability to support multiple graphics frameworks, specifically OpenGL, Vulkan and DirectX.

- Scribble2D depends on the following:
    - A Graphics API (OpenGL on release, Vulkan or DirectX in the future)
        - Tools or frameworks to use the chosen Graphics API, as follows:
            - OpenGL: GLFW and Glad
            - Vulkan: Unknown
            - DirectX: Unknown
    - glm: a performant C++ Mathematics library
    - premake: a build system using Lua scripting
    - spdlog: a performant and feature-rich C++ logging system.

3: Features

3.1 - Application

- To use Scribble2D, a user must instantiate a Scribble2D application object, which will be responsible for the life cycle of the application

3.2 - Window

- Scribble2D will support the creation and manipulation of windows with a valid render context

3.3 - Layer Stack

- Scribble2D will support the existence of the Layer Stack, a stack of renderable layers that will be drawn from the background to the foreground.

3.4 - Event System

- Scribble2D will contain a complete event system capable of handling input events and dispatching them between rendering layers on the layer stack.

3.5 - Texture and Shader management

- Scribble 2D will be able to load and manage textures and shaders within a file system that will be able to be loaded and applied to objects

3.6 - Sprite Renderer

- Scribble2D will be able to take a given sprite and render it onto the screen with the appropriate position, color, and transparency. Sprites rendered on a higher layer will cover up sprites existing on a lower layer.