# 2D Game Engine

**BETA**

# Software Design Document

CS-217 Abdelrahman Mohamed Soliman

ID:18200019

**TABLE OF CONTENTS**

1. [INTRODUCTION 2](#_bookmark0)
   1. [Purpose 2](#_bookmark1)
   2. [Scope 2](#_bookmark2)
   3. [Overview 2](#_bookmark3)
   4. [Reference Material 2](#_bookmark4)
2. [SYSTEM OVERVIEW 2](#_bookmark6)
3. [SYSTEM ARCHITECTURE 3](#_bookmark7)
   1. [Architectural Design 3](#_bookmark8)
   2. UML Diagram
4. [DATA DESIGN 3](#_bookmark11)
   1. [Data Description 3](#_bookmark12)

### INTRODUCTION

## Purpose

Beta is a 2D game engine that is designed to make 2D games:

1. Platformer games: Like Super Mario.
2. Endless runner games: Like Flappy bird.

## Scope

Beta is designed for indie game developers to make it easier for the to make 2D games without needing to care about importing libraries and dealing with the visualization, rendering, and sound settings and programming.

## Overview

Provide an overview of this document and its organization.

## Reference Material

External matrials used in Beta:

1-sdplog library: an open sourced library used for the logging system.

2-GLFW(Graphic Library framework): its an opengl library, and it is responsible for making windows

1. guerrilla sound: this is the library responsible of sound management
2. glad: open gl library that provides rendering APIs and shaders
3. premake2: used for generating projects automatically
4. imgui:

### SYSTEM OVERVIEW

Beta consists of two main parts:

1. Core: which is the Engine itself and this is the part that manages the whole rendering, logging, and event tracking, and it is a dynamic link library(DLL).
2. Client: and this the part that executes the code and works as the editor, and it generates the .exe file that runs the program(The game in our case).

### SYSTEM ARCHITECTURE

## Architectural Design

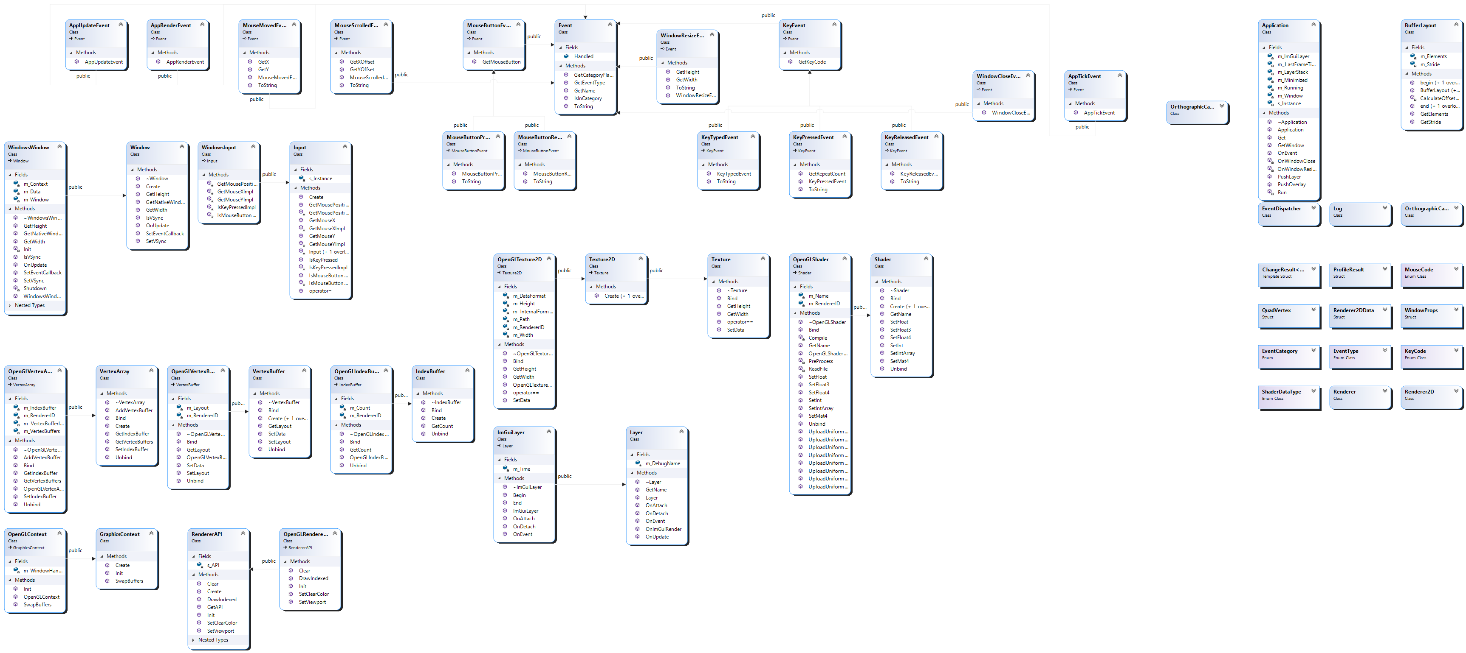
Beta Consists of two main parts:

1-Core(Beta): and this part contains the core header file that contains the macros of the project, and it is responsible of checking which OS the engine is running on, and the macros that manage the dllimport and dllexport for the engines classes to make it more dynamic.

Another class is the Application class that is responsible of linking between the engine and the client and it is the one that creates the application and manages the layers pushing and window generation. There is also the Log class that uses spdlog library. This class is the engine logging system. And the event class is the one that keeps track of every event in the engine while running and returns every event’s data to the engine using the EventCallBackFn(), and it can also return the event instance using EVENT\_CLASS\_TYPE() and EVENT\_CLASS\_CATEGORY() macros. The engine core also has a EntryPoint header that’s considered as the link between the engine and the operating system.

It also has the imGui and the rendering classes responsible of the graphics and the gui application in Beta engine.

## UML Diagram



### 4.DATA DESIGN

## Data Description

Profile name key codes mouse codes

Key state mouse state core logger

Client logger window title window height

Event type [enum event category[enum] mouse x offset

Mouse y offset shaders data type[enum] shaders data type size[enum]

Element buffer[struct] aspect ratio camera vector position

Camera rotation camera speed vertex array

Projection matrix .