Game Engine Programming

Overview

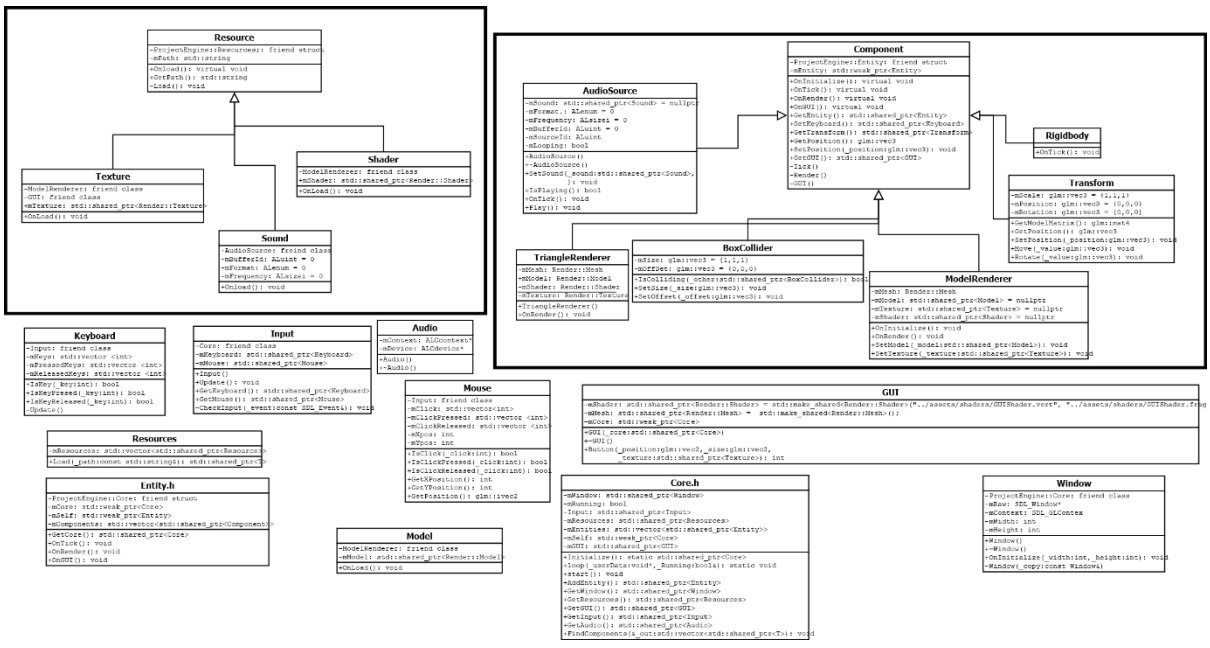
This report outlines the creation of my Game Engine (Project Engine). I will go over the key principles of a game engines such as scalability and modularity. The purpose of the project is to create a game engine which could be used by other people very easily and would be able to add any libaries and still be able to function. Along with the report is a functional game engine, demo, and source code.

Research

I have spent some time researching into other leading competitors the in the game engine market. Specifically, I looked at Unity which is also a uses an Entity component system architecture. [1] Unity has a lot of different features which I wanted to also implement into Project Engine.

Design

Project Engine is an Entity-component-system as other game engines such as Unity also use this same architecture. This means that if a person is to use my game engine and have experience with Untiy, Project Engine would feel familiar to the end user.



This UML diagram displays the modularity of the game engine I have creted. Each class is created for a specific purpose for the engine such as audio, collision, resources. The entity class associtaaes with component instances following the ECS architechture.

[1] Technologies, U. (n.d.). *ECS for Unity*. [online] unity.com. Available at: https://unity.com/ecs.

I created an Input structure for future development and intergrations such as controller and microphone. This is inline with my goal of the project being to create a modular system.

The system is design to be easily maintained. This is due to the clear seperation of each specific purpose meaning when a bug arises it is easier to find the specific files.

Development

Github was used as a version controller. This enabled me to recall changes which were made to the code in between sessions. It was also used to go back in time in code when debugging my program.

Also used Cmake to develop Project Engine. This is cross platfrom tool which uses compiler independent instructions to great the build files. The unique selling point to using Cmake is the cross-platform capability of the tool meaning that my engine will be able to run on other systems than the one I used to develop the engine.

Analysis

The engine can take Inputs to move entities and interact with the GUI which has been implemented. I have created a system which in future development will be able to take input from other devices such as a controller or microphone.

The engine can render many models and can produce sound. This system enables the user to be able change things like volume, channels etc. This allows the user to be creative with their own projects.

Potential improvements for the engine would be mesh collision, a camera system, animation, collision response, physics system and resource management and resource unloading. Adding these features would significantly enhance the capability and overall utility of Project Engine.

Overall, Project Engine is an easy to run game engine capable of generating models and producing sound. It has the basic functionality to create a game with minimal system requirements to do so. If more time was avaliable for the project, I would like to focus on a more advanced collision response system next.