UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

OpenGL Framework

author

Dragoş-Andrei Bobu

session: iunie, 2024

scientific coordinator

Conf. Dr. Varlan Cosmin

UNIVERSITATEA ”ALEXANDRU-IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

OpenGL Framework

Dragoş-Andrei Bobu

Sesiunea: iunie, 2024

Coordonator științific

Conf. Dr. Varlan Cosmin

Computer Software is just a bunch of tools orchestrated in just the right manner.

Similarly to this, a game engine is just a software framework of computer graphics components.

The Game Editor is what allows users to custom build new tools that work seemingly with the already existing SOLID engines.

Contents

Part I

Introduction

Introduction

Motivation

This project wished to evolve into a solid service with many endpoints that users can benefit from it's unique collection of components in order to build graphical experiences.

This edition's update journal will iterate over the implementation status of the first building blocks in the toolkit.

Introduction

The following chapters will present the components that were involved in building this Graphics Abstraction Layer.

Each component showcases at least one resource which is mentioned in the footer.

This implementation required extensive study of multiple computer science fields. Out of which i will name a few:

- Computer Graphics
- Game Development
- Machine Learning

Beside those, courses such as the following had made me an agile handyman that has his toolbox in order:

- oop, data structures and Software Engineering
- python programming, plp and Logics

Game Engines

In this section, we will inspect the criteria to which a piece of software needs to qualify in order to be considered part of the game engines set.

subsection*Natural Language Understanding

Wikipedia ″A game engine is a software framework primarily designed for the development of video games and generally includes relevant engines.″

Formal Definition

Let S be a piece of software, input for a function $isGameEngine$ that outputs a binary response.

$$\text{isGameEngine}(\text{Software } S) = \text{isFramework}(S) \wedge (\text{S::Components} \subseteq \text{GE::E})$$

where GameEngine::Engines represents the collection of all possible engines in all possible game engines, as such:

$$\text{GE::E} = \{\text{...Render}, \text{Script}, \text{Physics}, \text{AI}, \text{SFX}, \text{Robotics...}\}$$

Part II

Implementation

Implementation

Components

In this section, we will inspect each component present in the implementation.

Implementation

Part III

Bibliography

Bibliography

Literature

In the following, i will present all the literature i used in this research and it will be grouped based on what i learned from that said resource.

Mathematical Framework

″Mathematics For Game Developers” by AUTHOR

Rendering Engine

″Computer Graphics in C/C++” by AUTHOR

″opengl SUPERBIBLE” by AUTHOR

How does the screen know what the videoboard wants to render?

C++ Software Infrastructure

C++ by BJORN STROUPWAFFLE

Field Study

CS189 Stanford Computer Graphics Student Projects

CS201 Stanford Machine Learning Student Projects

Simulating Human Emunacra

Detective

This technology had been expanded further from research environments, this technology made it into production and there now exists services that allow access to fine-tuned ai models with special integration for popular use-cases.

(

How does the screen know what to display?)