

# **3D visual framework**

## **Functional Design Specification**

Jack Hulspas  
Version 1.0  
09/08/2023 (DD/MM/YYYY)

## Table of contents

<b>Table of contents</b>	<b>2</b>
<b>Introduction</b>	<b>2</b>
<b>Functionality</b>	<b>2</b>
<b>Features</b>	<b>3</b>
Core features	3
Future features	3

## Introduction

This framework is a rework and refactor of a previous project. The vision for this project is to make a 3D visual framework. The end-user will be able to use this framework to render 3D visuals in real time. The framework will render shadows and simulate simple collision and physics.

The end-user will be able to import their own textures and 3D models. The end-user will be able to create their own scene, with its own camera.

The framework will be designed with upgradability in mind. After the core features are implemented, new features could be implemented. Some of the features I plan to add after the core features are implemented are: Procedural generation, terrain generation, marching cubes, perlin/simplex noise, and custom file loaders.

The framework will be designed with a logical and expandable structure.

## Functionality

The intention of this project is to create a framework that can be used to render 3D objects in real time. The goal is to remove the low-level OpenGL code, allowing the end-user to render 3D scenes without any OpenGL knowledge.

The framework can be used to simulate scenes, create 3D video games, or any other use that requires the rendering of 3D objects.

# Features

## Core features

One of the features is an object and a texture loader, allowing the end-user to import visuals into the scene. To reduce the amount of calculations and math for the end-user, this framework will provide implementations for rendering, collision, physics, and shadow mapping.

The core features will be mainly focussed on simulating and rendering 3D objects.

## Future features

After the core features have been added, and this framework can be used to fully render 3D objects, new features will be added to expand the framework.

Some of the features that will be added in the future are about generation. For example, this framework will have an implementation of Perlin and Simplex noise, which can be used to procedurally generate various things. The framework will also have an implementation for terrain generation and marching cubes.

I also plan on creating a visual scene editor, which will allow the person to create a scene without the use of code. The core framework will require the end-user to create each object in the scene through code. However, I plan to create an editor which will allow the end-user to drag their objects into the scene.