3D visual framework

Project plan

Jack Hulspas
Version 1.0
10/08/2023 (DD/MM/YYYY)

Table of contents

Introduction

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.
The project will be well-planned out, and will have all the necessary documentation.

Research functional feasibility

Trello tasks

https://trello.com/b/fmrhelzh/framework-tasks

Assetlist

Scripts

- Runner script
- Camera class
- Renderer
- Scene
- Entity class
- Renderer class
- Material class
- Mesh class
- Model class
- Object loader class
- Texture loader class
- Sprite class
- Vector class
- Collision class
- 3D physics class

Version control

I will be using Git and Github for version control. I will commit new changes frequently and push them to a separate branch on Github. I will work on different branches, no code will be pushed directly to the master branch. Instead they will be pushed to another branch, before I will assign a merge request to the branch.

Backup strategy

I will store multiple copies of the project. Github will be used to have the project stored online. I also have 2 copies of the project on two different drives, which I will update frequently using version control.

Platform

The framework is designed to be used for Windows. However, the framework is able to run on different platforms, for example: Linux.
If you don't run the framework on Windows, depending on the platform, you might have to change the CMake file.

API / instructions

API / documentation

Documentation will be generated using Doxygen. If you wish to generate the documentation, make sure there is a directory "documentation" in the root directory. The doxyfile is located in the root folder, and will generate a "documentation.html" file inside the documentation folder.

Installation / build instructions

To install this project and start the demo, you can clone the project using Git.

1) Clone or fork the Github repository.
   https://github.com/JackHulspas/VisualizationFramework
2) Use the terminal, navigate to the root directory: **" VisualizationFramework/ "**
3) If there is no build directory, create one: **" mkdir build "**
4) Navigate to the build folder: **" cd build "**
5) Run the CMake command: **" cmake .. "**