

3D visual framework

Technical Design Document

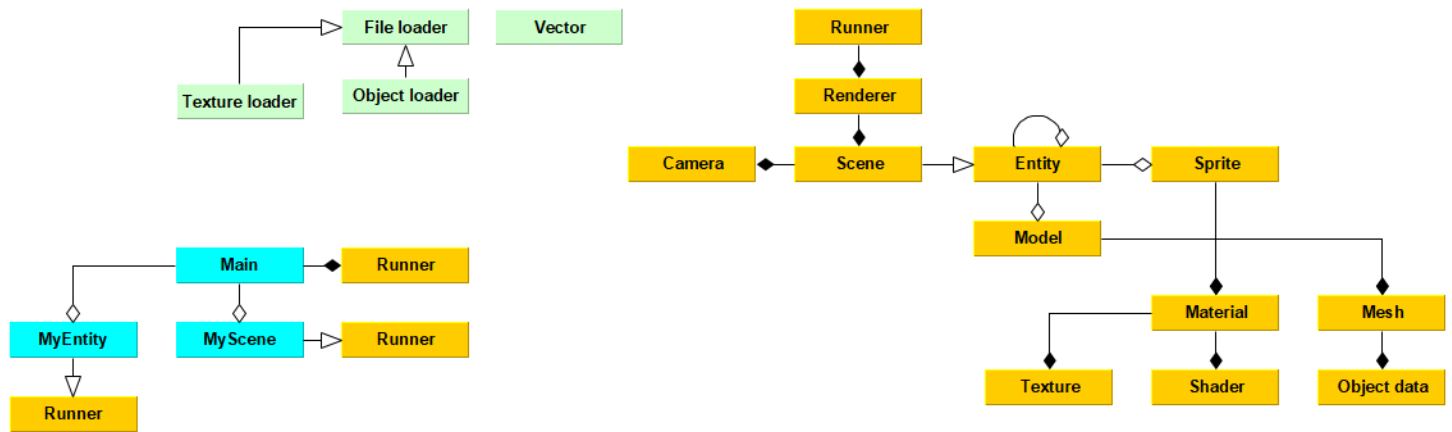
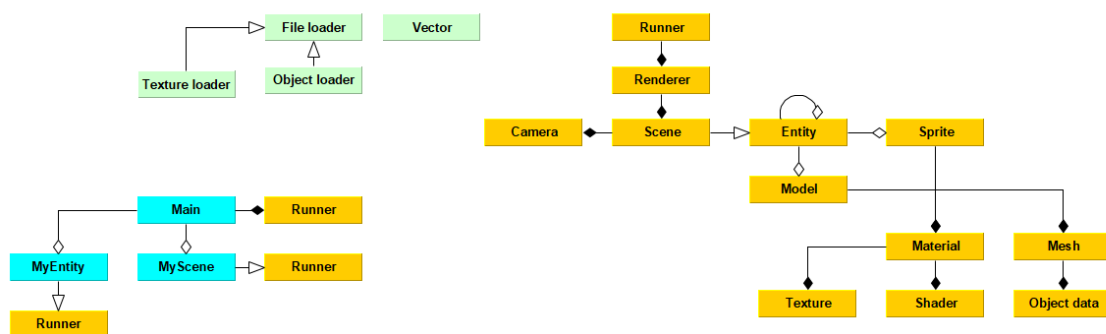


Table of contents

Table of contents	2
Class diagram	2
UML class diagram	2
Framework classes	3
Research functional feasibility	3
Trello tasks	3
Assetlist	4
Scripts	4
Assets structure	4
External	4
Demo	4
Framework	5
Assets specification	5
Classes	5
3D objects	5
Textures	5

Class diagram

UML class diagram



Framework classes

- Runner -> Runs the scene and the renderer
- Renderer -> Renders everything in the scene
- Entity -> The base class for most objects
- Scene -> Contains all the entity's.
- Camera -> Using the camera you can see all the entity's in a scene.
You can rotate or move around the scene using the camera.
- Model -> An entity with a mesh and material.
- Mesh -> The data for the visual 3D object.
- Material -> The material for an object, contains a shader and a texture.
- Texture -> The texture of a material or object.
- Shader -> Adds shading and shadows to objects in the scene.
- Collision -> Calculates collision between 2 objects.
- Physics -> Calculates physics for a 3D environment.
- Vector -> A vector class, which holds coordinates and calculations.

Research functional feasibility

Trello tasks

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

Assetlist

Scripts

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

Assets structure

External

- GLEW library
- GLFW library
- GLM library

Demo

- MyScene

Framework

- 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

Assets specification

Classes

Each class must be a .cpp file with a .h file as a header. The software must be modular, have high cohesion and will be as loosely coupled as possible. This ensures that it will be easier to test, maintain, and scale the framework.

3D objects

3D objects imported into the framework must be a .obj file. The .obj file must include normals and UVs. The .obj file must not import any materials. It must also triangulate the face. (Make sure those options are set in the software you use, IE: Blender).

Textures

The texture of a sprite must be a .tga file, and preferably the resolution must be a power of 2.

The texture of a model must be a .dds file, if you use a different format, the UV coordinates will not be correct. Once again, a resolution with the power of 2 is preferred.