

RTR 2024 Project - Deferred Renderer

This is the first submission of the project for Real-Time Rendering. So far, basic functionality has been implemented. Namely:

- Loading a scene ([Sponza Atrium](#))
- Basic lighting (Phong) which diffuse, normal and specular textures are being loaded. TBN transformations are being used for normals and alpha testing is performed to draw sprites correctly.
- Debug camera (manual and automatic predefined movement)
- **Deferred Shading** (1st advanced effect)
- **Screen-Space Ambient Occlusion** (2nd advanced effect)
- Unfortunately, I did not have time to implement the third effect, handling multiple lights (tiled rendering) on compute.
- GUI layer. This is used to display the effects more clearly, showcasing the intermediate buffers (G-Buffers) generated from the Geometry pass of shading in Deferred Mode, as well as the ambient occlusion map, and some basic controls, like moving the scene light around, toggling SSAO on/off and some SSAO intrinsic parameters (radius, intensity).

Installation

All necessary libraries are located in the ThirdParty folder. To build dependencies and project files, run the GenerateProjects.bat file, in the root folder. The project uses DirectX12 so it only runs on Windows environments.

Key Bindings

Key Bindings

- W A S D and mouse to move the camera around.
- Esc to close the app
- F1 mouse focus on window toggle on/off

Attention. The F1 toggle starts off which enables the camera to run on a predefined path. If the cursor focus gets toggled on, the camera is now manually-operated and cannot resume automatic movement. Also, because it reads positional and rotational data from a file, once the file reaches its end (about half a minute), it pauses in the last position specified by the file, it cannot resume and manual control is required.

Third Party Libraries

- DirectX 12 as the graphics API
- [DirectXTex](#) for loading images from disk
- [ImGui](#) for the graphical user interface layer
- [premake](#) for building project files
- [glm](#) was the selected math library
- [Assimp](#) for importing assets (specifically the .obj and .mtl files for the Sponza scene)

Specs

Tested on system with:

- NVIDIA GeForce GTX 1070 Ti
- Intel i7-8700K
- 16 GB RAM DDR4 @ 3200MHZ
- Windows 10