Documentation Submission 3

Group “Gestalt”

Implementation

Volumetric Light

We implemented volumetric light using a froxel-based approach. Based on submission 2, spotlight contributions were added to the scattering, the light frustum focus was improved, and synchronization issues leading to jittery fog were fixed.

Ray-traced Shadows

We implemented RTX shadows for the directional light using ray-queries.

Libraries

- fastgltf (https://github.com/spnda/fastgltf) - A lightweight, efficient library for parsing gltf, which allows loading complex 3D scenes for the demo
- fmt(https://github.com/fmtlib/fmt) - A modern formatting library used for debugging during the development process.
- glm (https://github.com/g-truc/glm) - A math library that provides vector and matrix operations. Used for calculating transformations, camera movement, lighting, and physics.
- imgui (https://github.com/ocornut/imgui) - A graphical user interface (GUI) library used to create a UI. Allows tweaking settings for rendering effects, and helps to debug on the
- Jolt Physics (https://github.com/jrouwe/JoltPhysics) - A real-time physics engine, although probably not used for this demo, is already part of the engine.
- meshoptimizer (https://github.com/zeux/meshoptimizer) - A library for generating meshlets and compressing geometry. This allows the demo to optimize rendering performance for complex models.
- SoLoud (https://github.com/jarikomppa/soloud) - An audio engine for playing the audio track
- SDL2 (https://github.com/libsdl-org/SDL) - Provides window and input management, making it easier to handle window creation and input capturing. Used for switching between automatic camera movements and manual control and controlling the UI.
- stb (https://github.com/nothings/stb) - A library used to load image files, such as textures fo the materials used in the demo scene.
- tracy (https://github.com/wolfpld/tracy) - A real-time, low-overhead profiler used to measur performance and ensure that the demo runs smoothly at the required 60 FPS.
- vk-bootstrap (https://github.com/charles-lunarg/vk-bootstrap) - A library for Vulkan initialization, managing device creation and validation layers.

- VulkanMemoryAllocator (https://github.com/GPUOpen-LibrariesAndSDKs/VulkanMemoryAllocator) - A library for memory allocations.
- volk (https://github.com/zeux/volk) - A lightweight loader for Vulkan extension functions

Graphics Card

We tested on an Nvidia RTX 3070 and RTX 3060

Visible Effects

The volumetric light is visible all over the scene and influenced by the sunlight and street lights. There is a height based fog, a distance based fog and a box fog which can be tweaked in the GUI menu: Window/Settings/Shading/Volumetric Light Settings. The sun can also be rotated using the Scene Graph menu on the right.

The RTX shadows are visible where the sun should cast shadows, they also move with the sun's rotation. Shadows can be turn off in Window/Settings/Shading/Shadow Mode

Additionally the engine implements:
Meshlet culling, frustum culling, PBR shading, SSAO, light adaptation using luminance histogram, deferred shading, skybox with atmosphere simulation, AgX tone mapping, color grading and a vignette filter.

Controls

F3 - toggles between Animation Camera and Free Camera and GUI

Free Camera Controls:
Right-click + Drag - rotates the camera
WASD - moves the camera
LSHIFT - moves the camera down
SPACE - moves the camera up
Pressing CTRL - speeds up movement

Light sources can be added via Edit / Add Light Source
Camera properties can be tweaked with Edit / Camera Editor

Configuration properties in bin/config.json
physicalDeviceIndex - used to select different GPUs, when more than 1 was found
useFullscreen - ignores window size and launches app in fullscreen
useValidationLayers - loads validation layers for vulkan
useVsync - changes present mode
windowHeight/windowWith - sets windows size when useFullscreen is false