Title

The Earth–Moon–Sun Dance

CSC317 Fall 2025 Showcase Extra Credit

Personal Information

Full Name: Zihan Zhao

UtorID: zhaoz188

Student Number: 1010128274

Assignment Augmented: Assignment 6 (Shader Pipeline)

Instructions

1. Configure the project (CMake)

From the Code/ directory:

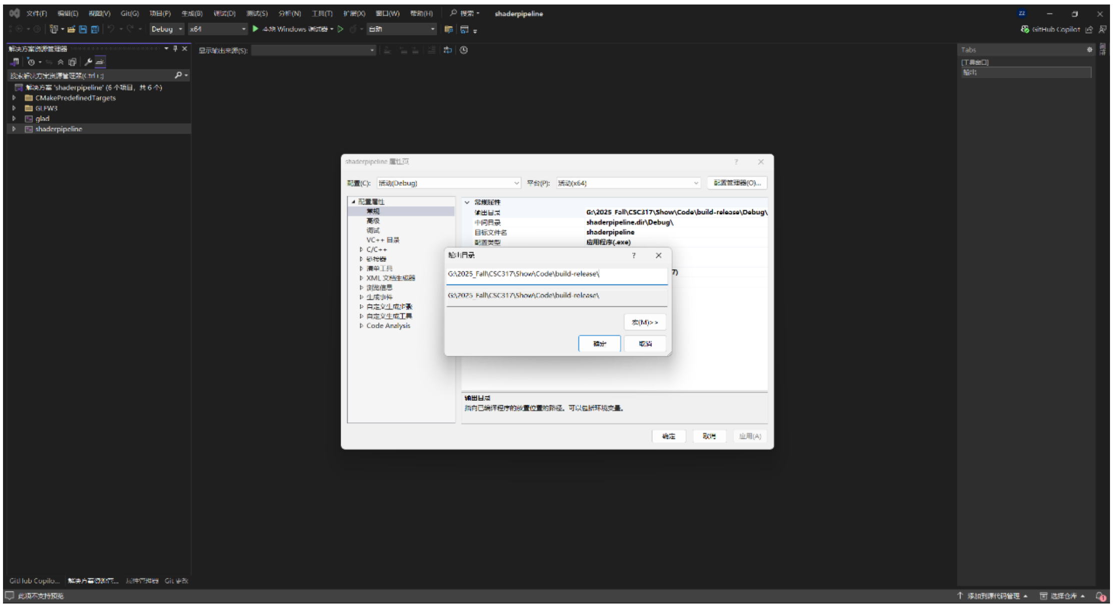
```
mkdir build-release
cd build-release
cmake ../ -DCMAKE_BUILD_TYPE=Release
```

2. Open in Visual Studio

Open shaderpipeline.sln inside build-release/

Change shaderpipeline target Output Directory to build-release/

Build the Release configuration.



shaderpipeline 属性页

配置(C): 活动(Debug)　平台(P): 活动(x64)

配置属性 > 常规

常规属性

输出目录	G:\2025_Fall\CSC317\Show\Code\build-release\Debug\
中间目录	shaderpipeline.dir\Debug\
目标文件名	shaderpipeline
配置类型	应用程序(.exe)

输出目录

G:\2025_Fall\CSC317\Show\Code\build-release\

G:\2025_Fall\CSC317\Show\Code\build-release\

输出目录

指向已编译程序的放置位置的路径。可以包括环境变量。

3. Run

Run the executable generated inside build-release/

Description

This showcase significantly extends Assignment 6 by enhancing three major components: the main application logic (main.cpp), the planet surface shader (planet.fs), the transformation system (model.glsl), and the procedural surface functions (bump_height.glsl).

1. main.cpp – Real-time animation, camera system, and shader integration
    - Added Earth–Moon–Sun orbital mechanics with nested model transformations.
    - Implemented a dynamic follow-Earth camera using a custom lookAt() function.
    - Added keyboard interactions (A, L, Z, H, R) for animation control and camera modes.
    - Passed sun_world_pos into shaders for physically-inspired lighting.
    - Integrated shader hot-reloading, automatically recompiling when files change.

2. planet.fs – Fully procedural Earth, Moon, Sun, and background starfield
    - Implemented a layered procedural Sun using animated Perlin noise, convection patterns, Fresnel glow, and emissive coloring.
    - Added multi-layer Earth shading: oceans, land, mountains, deserts, snow, polar caps, procedural clouds with soft shadows, and procedural night-time city lights.
    - Developed a procedural Moon surface using bump mapping and height-based color.
    - Added a procedural starfield background using multi-octave noise.
    - Unified all shading under a physically-based light model driven by sun_world_pos.

3. model.glsl – Orbital transformations for Sun, Earth, and Moon
    - Added complete transformation hierarchy: Sun, Earth orbit, Earth spin, Moon orbit, Moon tilt, Moon scale.
    - Tuned parameters for orbit radius, tilt, speed, and spin periods.
    - Added background sphere transformation for starfield rendering.

4. bump_height.glsl – Procedural surface generation
    - Implemented height-based terrain for Earth (ocean, coast, land, mountains, snow).
    - Added lunar crater-style bump height function for the Moon.
    - Provided height values used for dynamic shading, cloud shadows, night lights, and biome transitions.

Together, these components form a fully procedural real-time Earth–Moon–Sun system featuring physically-inspired lighting, detailed planetary rendering, and dynamic camera motion.

Acknowledgements

- libigl (icosahedron mesh generation)
- Ken Perlin's noise reference
- CSC317 Assignment 6 starter code
- GLFW, Eigen