References

Blinn, J.F. 1977. Models of light reflection for computer synthesized pictures. *SIGGRAPH Computer Graphics*. **11**(2), pp.192–198.

de Vries, J. 2020. *LearnOpenGL: Learn modern OpenGL graphics programming in a step-by-step fashion* [Online]. [Accessed 11 December 2025]. Available from: https://learnopengl.com.

Hughes, J.F., van Dam, A., McGuire, M., Sklar, D.F., Foley, J.D., Feiner, S.K. and Akeley, K. 2014. *Computer graphics: principles and practice*. 3rd ed. Upper Saddle River, NJ: Addison-Wesley Professional.

Lengyel, E. 2012. *Mathematics for 3D game programming and computer graphics*. 3rd ed. Boston: Course Technology PTR.

Appendix: Individual Contributions

The following table outlines the individual contributions of each group member to the codebase and the technical report. All members contributed substantially to their assigned modules.

Member	Contribution Details
Siyu Yu	**Task 1.1:** Implement fundamental 4×4 matrix operations including multiplication, rotation, translation, and perspective projection, validated through comprehensive unit tests. **Task 1.2:** Create a functional 3D renderer with perspective projection, first-person camera controls using keyboard and mouse input, and simplified directional lighting. **Task 1.3:** Add texture mapping capabilities by loading and applying orthophoto aerial imagery to the terrain mesh combined with lighting calculations. **Task 1.4:** Demonstrate geometry instancing by rendering two launchpad models at different sea locations using material colors and a separate shader program, showcasing efficient resource reuse in 3D graphics.
Yujie Feng	**Task 1.5:** Developed a procedural, hierarchical space vehicle model using geometric primitives, implementing affine transformations and inverse-transpose normal matrix calculations for correct shading. **Task 1.6:** Implemented the Blinn-Phong reflection model for multiple point lights, incorporating physically-based inverse-square distance attenuation and dynamic uniform updates. **Task 1.7:** Created a physics-based procedural animation system with a parametric curved trajectory, utilizing analytic derivatives and Rodrigues' rotation formula for orientation alignment. **Task 1.8:** Designed an advanced camera control system with a state machine architecture, implementing automated ”Follow” and ”Ground” tracking modes with seamless state transitions.
Haoyu Zhu	**Task 1.9:** Refactor the rendering logic into a unified function, implement left-right split-screen rendering, maintain two separate cameras with independent controls, and support split-screen toggling and window size adaptation. **Task 1.10:** Implement a particle system that uses a pre-allocated particle pool to emit particles from the engine, renders and handles depth in a specific way, and clarifies the implementation assumptions and limitations. **Task 1.11:** Build an immediate-mode UI system based on OpenGL + GLFW, include basic interactive components, specify the steps for adding new UI elements, and use a font atlas to optimize text rendering. **Task 1.12:** Implement GPU and CPU performance measurement functionality, analyze the test results, and identify performance bottlenecks and the system's real-time performance.