Project Proposal: Cross-Platform Game Engine Development

Abstract

This project involves developing a foundational cross-platform game engine from scratch using C++ and OpenGL. The engine will focus on essential features such as basic rendering, a simple physics system, user input handling, asset management, and UI integration. The goal is to create an efficient, modular engine that supports core functionalities, allowing developers to rapidly prototype and develop interactive games. While initially focused on essential components, the engine is designed with scalability in mind. It will be later developed to support physically-based rendering (PBR), deferred shading, high dynamic range (HDR) imaging, procedural terrain generation, real-time reflections, and realistic physics simulations—including rigid body dynamics, soft body interactions, fluid dynamics, and destructible environments. Built-in tools for procedural generation and networking will support the development of immersive open-world games in future expansions.

Project Overview

The game engine will provide foundational tools for building interactive gaming experiences, with an emphasis on modular architecture and cross-platform compatibility. The design will also facilitate parallelism and GPU-based computation to maximize performance across systems.

Game Engine Features

- **Rendering System:** Basic rendering of 2D/3D objects, with an architecture prepared for future PBR, HDR, and deferred shading support.
- **Physics System:** Simple collision handling and rigid body physics, designed to scale toward complex simulations.
- **User Input Handling:** Keyboard, mouse, and basic controller input support.
- **Asset Management:** Tools for loading and managing textures, models, and audio files.
- **UI Framework:** Custom OpenGL-based user interface for debugging and simple interactions.
- **Procedural Generation (Future):** Foundations for terrain generation, world population, and runtime content generation.
- **Multiplayer Networking (Future):** Extendable architecture to support multiplayer gameplay.

Team Members

- **Satrajit Ghosh**

Individual Responsibilities

As the sole developer, I will:

- Design and implement the rendering and physics subsystems.
- Build user input and asset loading systems.
- Develop a lightweight UI system integrated into the engine.

- Architect engine modules to support multithreading and GPU acceleration.
- Lay groundwork for procedural generation and networking features.
- Ensure portability across platforms with performance considerations.

Project Goals

Short-Term Goals (Semester Scope)

- Build foundational components: rendering, input, UI, asset loading, and basic physics.
- Optimize rendering and physics using multi-threading and GPU where applicable.
- Deliver a working demo with interactive elements and visual features.

Long-Term Goals (Master's Thesis Scope)

- Integrate advanced graphics techniques: PBR, deferred shading, HDR, real-time reflections.
- Implement terrain generation and procedural world building tools.
- Expand the physics system with soft bodies, fluids, and destructible environments.
- Add generative AI for NPCs and gameplay behavior.
- Implement a modular networking layer for multiplayer support.
- Leverage GPU parallelization and cloud offloading to scale AI and simulation workloads.

Conclusion

This game engine project will begin with a foundational implementation and grow into a scalable and modular system for high-performance, AI-augmented game development. With a focus on core engine features and forward-looking architecture, the engine will serve both short-term development and long-term research objectives in real-time graphics and intelligent interactive systems.

References

[1] Gregory, J. (2023). *Game Engine Architecture*. Retrieved from https://www.gameenginebook.com/

[2] Lague, S. (2023). *Sebastian Lague YouTube Channel*. Retrieved from https://www.youtube.com/@SebastianLague/playlists

[3] Chernikov, Y. (2023). *The Cherno YouTube Channel*. Retrieved from https://www.youtube.com/@TheCherno/videos

[4] Lisyarus. (2022). *So, You Want to Make a Game Engine?*. Retrieved from https://lisyarus.github.io/blog/posts/so-you-want-to-make-a-game-engine.html

[5] Reddit Community. (2023). *r/gameenginedevs*. Retrieved from https://www.reddit.com/r/gameenginedevs/