

Master thesis : Minecraft Clone in Rust and Vulkan

Auteur : Leclipteur, Henry

Promoteur(s) : Mathy, Laurent

Faculté : Faculté des Sciences appliquées

Diplôme : Master en sciences informatiques, à finalité spécialisée en "computer systems security"

Année académique : 2024-2025

URI/URL : <http://hdl.handle.net/2268.2/22449>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

Minecraft Clone in Rust and Vulkan

Author: Leclipteur Henry

Supervisor: Prof. Mathy Laurent

Academic year: 2024-2025

Thesis Summary

This master's thesis describes the development of a Minecraft clone using the Rust programming language and the Vulkan graphics API. The primary objective was to establish a foundational framework for a voxel-based game, allowing students to expand upon it for their personal projects or master's thesis. My personal goal was to explore and understand the various mechanisms involved in game development, ranging from 3D rendering to game physics.

The decision to use Rust and Vulkan was guided by my supervisor, Professor Laurent Mathy. Rust offers significant advantages in game development, particularly in its robust handling of memory safety, which eliminates many of the most challenging bugs associated with memory management. Vulkan, as a low-level graphics API, provided valuable insight into the intricacies of the rendering pipeline, as it requires explicit control over all stages of the process.

The project culminated in a functional game composed of four distinct modules:

1. **Renderer:** The rendering system built in rust and using the Vulkan API for efficient graphics processing.
2. **Game core:** Responsible for modeling the game world and mechanics.
3. **Main:** Coordinates the overall game execution.
4. **Draw element:** Facilitates interaction between crates to maintain modularity.

The most significant challenges of this project was developing the rendering system using Vulkan. As a low-level graphics API, Vulkan requires precise specification of every aspect of the rendering process, presenting a steep learning curve for beginners. Despite its complexity, I successfully implemented a rendering system capable of rendering up to 225 chunks around the player while maintaining a stable framerate of 144 FPS.

Achieving this performance required multiple optimizations, including minimizing delays during chunk loading and implementing a greedy meshing algorithm to reduce rendering overhead.

The resulting Minecraft clone, while rudimentary, is fully functional. Players can navigate a procedurally generated world and interact with the environment by placing and removing blocks. However, the current implementation is in its early stages and lacks several advanced features such as animations, multiple entities, and multiplayer functionality.

Future improvements will focus on further optimizing the rendering system and expanding the game's feature set with additions such as sound, crafting mechanics, animations, and improved user interfaces. This project serves as a foundational resource for anyone interested in learning about voxel-based game development.

The code and documentation are available on my GitLab, where they are presented in a clear and easily understandable manner. This provides students and developers with a straightforward framework for exploring game design and graphics programming.