SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

Faculty of Informatics and Information Technologies

Rust course

Final project

MineGit

Made by:

Roman Hanushchak

Vladimir Riazantsev

Github:

Project: MineGit
Roman Hanushchak: ROmanGanushchak
Vladimir Riazantsev: BRUH1284

Introduction

The project will provide a version control system, similar to Git, for Minecraft worlds.

MineGit operates independently of the game, working directly with Minecraft world files to enable efficient versioning and backups.

The program should allow players to easily back up their worlds and restore them through the CLI. Git or similar existing programs will not be used for this functionality.

During restoration, users will be able to specify specific regions (parts of the world) to be restored. By default, all regions will be restored.

Requirements

- Not damage the data of the user's world.
- Support committing current changes.
- Provide rollback capabilities to restore previous versions.
- Allow users to specify specific regions to be restored or commited.
- Be able to create new commits from the rolled back version.
- Provide a command-line interface (CLI) for managing world versions.

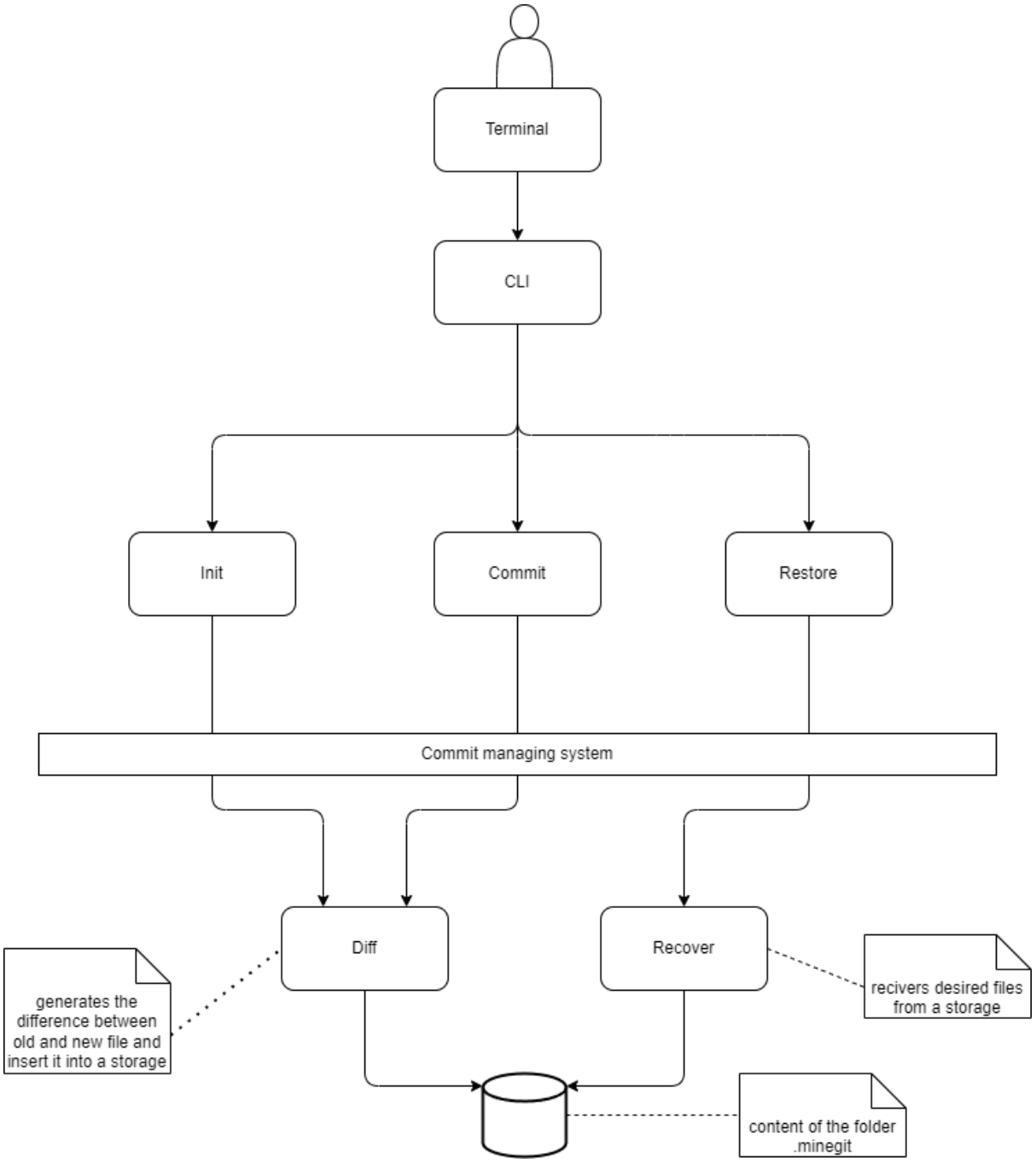
Dependencies

- bitcode: The bitcode dependency is used for encoding and decoding Commit data. It was chosen based on results from rust_serialization_benchmark, where it demonstrated the fastest serialization and deserialization speeds. Additionally, it produced

compact serialized output that compressed especially well with zstd, which we use to compress file descriptions within commits.

- bytemuck: The bytemuck dependency is used for safely converting data structures to and from raw byte slices when a static, predictable memory layout is required-for example, when writing fixed-size entries.
- chrono: The chrono crate is used for working with timestamps and converting them into human-readable date and time formats.
- clap: The clap (Command Line Argument Parser) crate is used to define and parse command-line arguments in a declarative, user-friendly way.
- glob: The glob crate is used to define patterns for matching files or directories. In this project, it helps implement an IgnoreFilter that allows certain files or directories to be excluded from commit based on patterns in ignore file, similar to .gitignore behavior.
- sha2: The sha2 crate is used to calculate and compare file hashes, which helps determine whether a file has changed between commits.
- zstd: The zstd crate is used for compressing commit data, providing high compression ratios and fast decompression speeds.
- tokio: The tokio crate is used to handle asynchronous tasks, allowing for efficient parallel execution of file comparison and commit operations.
- byteorder: this library is used to read and write primitive types from/to a file. Is used for DiffCommand and SnapshotHeader serialization/deserialization
- divsufsort: is used for suffix array generation during file difference generation. Although it is not the fastest library asymptotically, it proved to be the most efficient for our use cases during testing.

Design diagram



Design choices

The whole system uses the concept of diff files, instead of restoring files each time they are saved, the program generates diff files that store changes that need to be applied to the original file to restore the new file.
Diff uses 2 commands Insert and Copy
Diff generation - for diff generation Bsdiff algorithm was used, its main advantage is smaller diff file size in a cost of a longer diff generation time. It uses a suffix tree to find a longest common substring between 2 byte arrays, for this chunk copy is applied.

Evaluation

Overall we were able to implement the desired functionality of our project. But there is a room for improvements, we were not able to fully capitalize the .mca format for faster commits, it is mostly due to both contributors having a side subject that took most of the time in the final weeks. That said all required functions were implemented and worked correctly.

Rust seemed like a different programming language, it does not try to be like others and has some cool things in it, the most annoying thing for me personally was error handling that forced me to manually find the cause of an error or unwrap every error.