Netsketch: A Collaborative Whiteboard

Assignment Report

Giorgio Grigolo

Contents

1 Introduction

This implementation of Netsketch, a collaborative whiteboard, is built entirely in Rust, a statically typed, memory-safe, idiomatic, systems programming language.

1.1 Project Structure

Netsketch was divided into three main components: the *server*, the *client*, and a *shared library*. The server is responsible for most of the business logic, such as managing the state of the whiteboard and broadcasting changes to all connected clients. The client is a graphical user interface that allows users to draw on the whiteboard and see the changes made by other users. The shared library contains the data structures and algorithms used by both the server and the client.

To conform with the Rust ecosystem's conventions, the above components are manifested as a Rust workspace, with the server and the client as separate crates, and the shared library as a library crate, initialized by the `cargo init --lib` command.

1.2 Libraries Used

A considerable effort has been made to use as few external libraries (or *crates*) as possible. The only crates used are

- `clap` for the command-line interfaces of the server and the client,
- thiserror for better error handling,
- `bincode` for encoding and decoding data structures into and from bytes,
- `macroquad` for the client's graphical user interface, and
- `tracing` for server side logging.

All other functionality has been implemented from scratch, or was built using the standard library `std`.

2 System Design

2.1 Data Structures

2.2 Shared Library