

# CS 410P - Rust Programming - Project Proposal, A Rusty Musicbox (Fall 2023)

October 27, 2023

## 1 CS 410P: Rust Programming, Fall 2023 - Course Project Proposal

Dan Jang, yeh2@pdx.edu - 10/27/2023

### 1.1 Project Title:

*A Rusty Musicbox*

#### 1.1.1 Team Member(s):

Dan J.

**GitHub Repository URL:** <https://github.com/DebuggerDan/a-rusty-musicbox>

**Project Topic Area:** Music generation, playback, and simulation!

**Project Vision:** Creating a Rust-based program that will feature music & audio generation, playback, & simulation.

Specifically, *A Rusty Musicbox* will feature hopefully one or more instruments for simulation, at least a virtual piano or a MIDI-emulation feature, playback of generated music, & if time allows / if feasible, a simple GUI for user-interactivity.

The following features are currently in mind for this project:

- I.) Some sort of instrument simulation, e.g., a virtual piano.
- II.) MIDI-file processing or parsing used in the project.
- III.) Music generation.
- IV.) If feasible / time allows, interactivity, ideally, for controlling music generation.

I hope to make this project feature, hopefully, more than just a MIDI/Piano player, hence, it not being called *A Rusty Piano*, but try to apply my knowledge & skillsets learned from our wonderful Spring '22 class, *CS 410P: Music, Sound, & Computers*, and specifically, taking concepts learned from building our then-group's [M.A.N.G.O. Project](#) (*Musical Analysis Note Generation Operator*)

- where our group had built a *Python*-based music generator, that 'interpreted' Python code to generate a specific, generated music file.

However, since we do not want to simply repeat previous course projects, in this Rust project for Fall 2023, I will instead try to focus on making a virtual musicbox program, that generates music more based in the traditional sense of playing music, e.g., instrument(s), MIDI parsing/processing, interactive music generation, etc.

**Possible Project Issues:** While I do think an interactive functionality through a GUI/UI may be key in making *A Rusty Musicbox* seem more fun to use & seems 'right to do' for a cool musicbox program, I have read that *Rust* may be [slightly more difficult](#) comparatively in implementing GUIs.

Thus, in regards to interactivity or with the possible GUI/UI features, I may have to resort to switching focus to more on the library crate aspect more than making the GUI/UI more refined/cooler, or hopefully not, where I may have to resort to purely a library crate-based project - in that case, at least a CLI-based example for driving the program may still be feasible, e.g., possibly taking in live keyboard input for a simulated piano feature, parsing MIDI files for virtual piano (or other instruments) music playback, etc.

Otherwise, I think the project should be feasible either as a library crate or one bundled with a cool/simple GUI/UI, where I want to focus on usability or the element of fun from the functionalities of the project, and hopefully, make a neat Rust course project.