Final Project Proposal

The goal of the final project is for you to gain experience with developing Rust outside the guardrails of the projects we've seen so far. Ideally, you would build something that excites you and requires you to deal with a new part of the language or deal more deeply with a part of the language we've explored already. For ideas, look at the last slide of the most recent lecture. If you are completing a senior design project or similar capstone work, you're welcome to combine that with this final project so long as you still incorporate Rust. **The final project is due before the final lecture (December 2nd). You will share your results in a presentation during that lecture.**

1. Provide a brief overview of your project.

2. What parts of the Rust language will you use that we either haven't covered in class or haven't covered in as much detail as will be required for your project?

3. What are the main challenges you foresee encountering?

4. What is the minimum project you would need to produce in order to meet your goals?

5. If your project ends up being easier than anticipated, what are some stretch goals you could attempt as extensions?

6. What resources will you need to complete your project? This could be many things, from a dataset to evaluate your implementation on, to libraries to provide code for you, to cluster computing to run large programs. If there's something you need but can't find or don't have access to, the course staff may be able to help.

	What is it?	Meets expectations	Example
Code	The code that you submit on Gradescope	• Code shows mastery of topics covered in lecture • Code shows knowledge gained independently on topics or crates not covered in previous assignments	• Not making mistakes on topics covered in class such as using an `Arc` when a `Rc` would suffice • Using some new feature of the toolchain like compiling to web assembly/smart contract/ARM/python bindings • Using a new feature of the language like unsafe, async, advanced traits, macros, etc. • Using a complicated crate for things like game programming, auto-parallelization, serialization, networking, or graphics
Evaluation	The README of your submission	• README describes each feature implemented • Quantitative or qualitative results are clearly displayed with instructions demonstrating how to reproduce the results	• (For a command line tool) example invocations and expected output • (For a data science package) Instructions for downloading a dataset as well as expected results on that dataset • (For a game) example screenshots and guide to build and run • (For project with non-cargo dependencies) a docker container with necessary dependencies • (For non-trivially reproducible projects e.g. hardware) images or videos demonstrating projects in action
Presentation	The presentation you give to the class (approximately 10 minutes)	• Presentation clearly describes what you accomplished and the results you obtained • Presentation teaches fellow students about a topic not covered in other projects • Presentation reflects on project completion	• Showing some of your results (figures, images, etc.) • Giving a brief overview of a non-trivial crate that you used • Giving an example of how to use an advanced language feature • **Describing what you found upsetting/interesting/enjoyable about using Rust on a larger project**