Project Instructions (50.051 Programming Language Concepts, Y2025)

Simon Perrault, Matthieu De Mari

1. Introduction

In this project, you may work on a theme of your own choice (assuming it fulfils a minimal list of requirements detailed below). You will have to design your own custom system. The system should have a use, be it for productivity, data processing or even gaming.

Please assume that this project is to be submitted to a serious client (played by us professors) and take this project as an opportunity to work on your project delivery skills. More specifically, pay attention to the way you code and document your code.

Your submission will have to be uploaded for grading. Please only submit your code as a Zip file containing your project and a Makefile on eDimension. We also **more than strongly advise** you to upload your final project on Github, to enrich your personal portfolio of projects.

Finally, we expect your group to present the project during one of the last two sessions of Week 13, time to be confirmed closer to the date.

2. TLDR

Groups of up to 4 people, we will not allow people to work alone.

Free choice on project proposal.

Custom project proposal deadline and groups formation before 24 February, 23.59.

Use link for group formation: https://docs.google.com/spreadsheets/d/1Wg9DKszC-k40gWThEZaLwOEe--9RdQC-PRPHODaltS8/edit?usp=sharing

Submission deadline: 21 April, 23.59.

3. Objectives and guidelines for project proposal

You will need to submit the project proposal via email to me (simon_perraulti@sutd.edu.sg) by March 1st 23.59 for approval.

The proposal should be a small PDF file, containing a brief description of the following elements:

- Topic, problem to be investigated,
- Expected inputs and outputs,
- Team members,
- What you are going to deliver.

Project requirements: The constraints we have decided to give for this project is the following:

1. *Your system should process files as an input and output files at the end/during the process*
2. You should create your own parser. The parser should be robust and be able to detect incorrect input
3. Overall, the project should combine the main aspects taught in this class:
    a. Clean C programming (the whole project is in C, and should *ideally* compile with the usual -Wall -Werror -ansi -pedantic)
    b. File reading/writing (both text and binary)
    c. State machines
    d. Parser
    e. No external libraries, except for specific cases below
4. The project will be evaluated on the quality of the code, and the complexity of the four features above (e.g. more complex parser will score more points). You may also include multiple, simpler parsers/state machines

Here are some examples of files that could be used as input or output:

- XML
- LaTeX
- CSV
- JSON
- Image (if you choose to use these format, contact Simon to discuss the use of libraries)
- Audio files (if you choose to use these format, contact Simon to discuss the use of libraries)

For most of these examples, you could decide to go with a similar format with limited features and write a parser yourself, which would satisfy 2 constraints at once.

State machines could be used for:

- Parsing a text file
- Keeping track of a combat system (e.g. in a RPG)

Apart from these constraints, we are rather open to crazy topics!

4. List of additional expectations for projects

As mentioned before, you are expected to deliver a project, as if us professors were professional clients. As such, good practices will be critical, for instance:

- Make sure the code can compile and run on Windows, Linux and Mac (at least 2 of them)
- Your project should have a purpose (process data, apply filters to photos, a fully fledged text-based game, etc…)
- Your project should be well written (code and report) and documented
- Your report should contain everything we need to know to run your code
- Put in your submission the group members and their contribution to the project.

5. Project Delivery

Delivery details

Groups of up to 4 people, and we will not allow people to work alone.

Free theme on project proposal, only requirement is that it should meet the constraints outlined above.

Custom project proposal deadline and groups formation before 1 March, 23.59.

Submission deadline: 21 April, 23.59.

Final report

Your final report shall be submitted in a PDF format. Note: there are two items on eDimension, one of the code, one for the report. Please submit them separately.

The report should present the project, and how the constraints given in this document are satisfied with your project. You may also want to explain how your system works, what it does.

You then should explain the development process, with the difficulties you encountered, how the work was split among yourselves and how you made sure that the code worked on the computer of each member.

There is no page limit (max or min), the goal is that someone who does not know about the project is able to understand:

- The context
- How you worked with the constraints
- Challenges
- What each of you did
- What you learned from the project.

Code

You code may consist of folders with `.c` and `.h` files (no C++) as well as example data (if relevant), and a Makefile. A README file should be included to explicitly mention what needs to be done to run your code (commands to run if any, etc.)

Properly documenting your code files would be much appreciated, and you should practice it anyway (they are good practice!).

Project delivery

We strongly advise to upload your submission (code + PDFs) on a Github repository. You can then submit the link to your PUBLIC Github repository, during your submission on eDimension, as part of the report.

Your Github repository for this project should contain your PDF report, your DOCUMENTED code. It should also contain directions showing the required libraries and steps needed to run the project. If you are using an authorized library that may only work on one system, this should be clarified.

This is essential, for reproducibility and transparency reasons and something that we, researchers, must do constantly for our papers. In addition, using a github repository could help in case there is a conflict in the group, i.e. so we can determine who contributed and who did not.

Important note: you are nearing the end of your curriculum at SUTD, gathering your projects, and uploading them on your personal Github is much appreciated by recruiters as it allows them to immediately identify what your coding capabilities are. If you have not done so yet, please consider starting your project portfolio on Github, and/or creating a simple web page to present your portfolio!

Project presentation

Finally, we expect you to present the project during one of the sessions of Week 13. A small demo, along with some slides or a small video would be appreciated. Time and details to be confirmed closer to the date.

Grading rubric

Project originality, quality of the report, quality of the code and presentation will be the main components of the final grade given to this project.