Final Project Guidelines

Purpose

The purpose of the final project is to give you an opportunity to:

- put into practice the ideas you've been learning this term;
- hone your design and implementation skills;
- experience the construction of a non-trivial application;
- work in a team and develop your leadership and collaboration skills;
- and have fun!

Key Deliverables and Dates

The final project schedule (available in a separate document on Stellar) summarizes the key deliverables and dates.

Division of Labor

You can divide the work amongst team members as you please, with two caveats: (a) the amount of work per team member must be roughly the same, and (b) each team member must participate in all software development activities (ie, design, coding, testing, documenting, etc).

Mentoring Meetings

Each team will be assigned two teaching assistants, one as a lead mentor, and the other as a consultant mentor. You will have weekly one-hour meetings with the two mentors, at times to be mutually agreed upon. Each team is responsible for producing agendas, meeting notes and progress reports, as specified in the project checklist. For each meeting, the team should have prepared in advance a report of progress in the previous week (in relation to the milestones of the project plan and particular items planned for the week in the last meeting), and should also bring an agenda consisting of a list of issues to be discussed in the meeting. During the meeting, a member of the team should take brief notes. After the meeting, the agenda should be augmented with the meeting notes and any particular outcomes (such as decisions made, alterations to the project plan, etc). The agenda and progress report should be checked in to the project repository before the meeting and the meeting minutes shortly after.

Project Pitch

Near the start of the project, your team will present a very brief pitch to the whole class, explaining why you're doing the project, what the key concepts and features are, and how you plan to mitigate the risks you anticipate.

The length of the presentation will be determined by the number of teams, but is expected to be between 3 and 5 minutes.

Minimum Viable Product (MVP)

Early on in the project, your team will complete and demonstrate to the class a minimum viable product (MVP). The purpose of the MVP is to get some code up and running to ensure that your development process is working and you can integrate the work of all the team members, and also as a proof of concept to try out ideas when you still have time for a course correction. You can also use the MVP as a way to mitigate risk by starting with an implementation of the feature you're most concerned about.

The MVP is to be specified in the project plan as a subset of the features and concepts described in more detail in the design document. You do not need to do a separate, standalone design for the MVP, but should be able to specify it just by listing concepts or features to be included or excluded.

You are not required to provide specs and test cases for your MVP. This is to allow you to use the MVP as an opportunity to conduct lightweight experiments with different implementation ideas. That said, you should be very wary of incurring "technical debt": if you write lots of poorly commented and sloppy code, it will be very painful to fix it later. So at the very least, make use of TODO and FIXME comments to mark areas of code that will need to be revisited.

Teamwork Plan and Team Contract

Writing the teamwork plan gives you a chance as a team to think about your goals for the project, what challenges you'll face, and what you'll do if things don't go according to plan. Don't waste the chance by treating it as a chore to be dealt with perfunctorily. The team contract should be the result of an open and candid discussion amongst team members about what your individual aspirations and commitments are; if you do this thoughtfully, it will reduce the chance that your project will be derailed by misunderstandings and disagreements.

Peer reviews

At the end of your project, you'll write a reflection as a team on the technical decisions (under Design) and on the team planning aspect (under Teamwork). You'll also each individually write a brief peer review of each of the other team members, each about 50 words long, giving some constructive comments on how that member might improve their teamwork and technical skills. These comments should be shared openly amongst the team and submitted in the final checkin. They will not be used to adjust individual grades, but if the comments are well done, the team as a whole will receive credit accordingly.

Grading and Awards

The grading breakdown of the various deliverables is given in the final project schedule. In general, all members of a team will receive the same grade, unless there are extenuating circumstances. The programming portion of the grade will include an evaluation of the functionality of the application (which is not listed as a separate item in the project checklist, but which is obviously important).

Awards will be given for the best projects; nominations are not required.