Team Contract

A team contract is an agreement between you and your teammates about how your team will operate — a set of conventions that you plan to abide by. The questions below will help you consider what might go into your team contract. You should also think back to good or bad aspects of team project experiences you've already had.
Below are some questions to consider. Your contract doesn't have to answer all the questions below, but must answer the boldfaced questions. Focus on the issues that your team considers most important.

Goals

- What are the goals of the team?
  - Kick ass - Do the basics of the project extremely well
  - If time, go above and beyond basic requirements with cool features
  - Develop and execute proper software development habits
- What are your personal goals for this assignment?
  - cjford - Learn how to code "professionally" (i.e. proper coding practices for working in teams), develop vast testing suites that have great coverage, think things through before coding
  - afreel - Build a solid piece of software that works efficiently and is organized.
  - jhaimson - Practice good software development principles, use this as an opportunity to start a personal project to create chorale and baroque music generation software
- What kind of obstacles might you encounter in reaching your goals?
  - Time constraints
  - Difficulty of project
  - Interpersonal difficulties
  - Intrapersonal difficulties
- What happens if all of you decide you want to get an A grade, but because of time constraints, one person decides that a B will be acceptable?
  - Other people on the team step up if they can and the person who ran out of time feels bad about himself
- Is it acceptable for one or two team members to do more work than the others in order to get the team an A?
  - Although significantly less than ideal, yes.
  - Person who does less work buys dinner for the team to celebrate the A

Meeting Norms

- Do you have a preference for when meetings will be held? Do you have a preference for where they should be held?
  - At the convenience of all team members (evenings and early afternoons)
- **How will you use the in-class time?**
  - Longer meetings to work through specific problems, integrate components and plan out the project
- How often do you think the team will need to meet outside of class? How long do you anticipate

meetings will be?
  - Once a day “Scrum” sessions - 15 minutes
    - Strict timeline
    - Talk about what has been accomplished since last scrum, what will get accomplished by next meeting, problems faced
  - Three times a week in class meetings
- How will you record and distribute the minutes and action lists produced by each meeting?
  - Google Docs with PDF’s in git repository

Work Norms

- How much time per week do you anticipate it will take to make the project successful?
  - 25-30 hours
- How will work be distributed?
  - Based on skillset, availablility
- How will deadlines be set?
  - Agreed upon during scrums/meetings
- How will you decide who should do which tasks?
  - We will take a look at everyone’s strengths and weeknesses and figure out who would be best suited for what given their skillset and familiarity with the material.
- **Where will you record who is responsible for which tasks?**
  - In Trello
- What will happen if someone does not follow through on a commitment (e.g., missing a deadline, not showing up to meetings)?
  - Lateness or inability to meet a deadline will be communicated as soon as possible. Others will work to pick up slack, help with work, etc.
- How will the work be reviewed?
  - By way of using the code reviewing features on github
- What happens if people have different opinions on the quality of the work?
  - Talk it out. Distill exactly what differs the opinions and figure out how large these differences are and how we can go about resolving them.
- What will you do if one or more team members are not doing their share of the work?
  - Point out the effects of not doing one’s part and work with that individual to get them back on track. Figure out the cause of this lack of effort (laziness, other work, personal problems) and work with other members of the team to develop and then execute a plan to deal with the problem.
- How will you deal with different work habits of individual team members (e.g., some people like to get assignments done as early as possible; others like to work under the pressure of a deadline)?
  - As best as possible, find happy mediums between different people’s work styles. Make coding as modular as possible to allow people to work when best suits them.

Decision Making

- Do you need consensus (100% approval of all team members) before making a decision?

    - On large decisions, absolutely. When it comes to a small coding decision as to how to go about doing something, 100% approval is not necessary.
- What will you do if one of you fixates on a particular idea?
    - Have a conversation as to the merit of the idea and figure out if it makes good sense to implement the idea or not.