Grading Rubric for CSSE 120 Project – Fall term, 2013-2014

Each team member normally gets the same score for Process and Green Features. For the remaining points, each team member earns them individually (but team members can and should cooperate on all parts of the project). ***Additionally, your score can be adjusted (either up or down) per your individual performance*** – how much you contributed to the project and your "team citizenship".

Process: ***15%*** - 5% per sprint, graded soon after the sprint ends.

Full credit for the sprint if, at the end of the sprint:

- Each team member's task list is up-to-date and in the correct format. ***If even one team member's task list is incorrect, no team member gets credit*** unless you can show me email(s) from another team member reminding this member to update his/her list.
- Feature list is up-to-date for just-completed sprint and new sprint. (But Sprint 3 has no "new sprint", of course.) Completed feature list from previous sprint matches the reality of what is actually working.

During the sprint: Students should keep task lists up-to-date. We reserve the right to check for this ***during*** sprints as well as at the end of sprints, and to deduct points on an individual basis as appropriate.

Green features: 15%

- Feature 1 (basic and advanced): 5% (3 basic, 2advanced)
- Feature 2 (basic and advanced): 10% (5 basic, 5 advanced)

Blue features (graded separately for each team member)*

25% (15 basic, 10 advanced)

Yellow features (graded separately for each team member)*

30% (15 basic, 15 advanced)

Additional features: 15%

Your instructor will judge the point value of each feature and which team member(s) get credit for it after demonstrations and conversations with the team. Two or three features (each done "reasonably") taken from numbers 9-17 should generally suffice for full credit; but you should okay the specific things you have in mind with your instructor.

Adjustments to the total score (up and down) may be made based on "team citizenship", code quality and the degree to which you understand the code.

Understand the code: Do you understand all of "your" code in ***complete detail***? Do you understand the general outline of all of your teammates' code?

Team citizenship: Did you attend all team meetings? Participate throughout the project? Contribute fully to "shared" code? Coordinate your work with teammates? Complete the mandatory "peer reviews" at the middle and end of the project? And so forth?

Code Quality: Is your code high quality code, per what we have modeled all term in the code we have supplied to you? In particular:

- Variable and function/method names that clearly indicate how the variables/functions/methods are used. Examples:
  - **`button2`** is a poor variable name; **`go_until_wall_button`** is a much better name.
  - **`go_until_wall`** is an excellent name for a function but a poor name for a button.
- A ***docstring*** for each function or method, which describes what the function/method does or returns, what types of parameters it expects and expected preconditions of arguments (e.g., "***`way_points is a nonempty list of Points`***").
- No magic numbers in your code; used named constants instead.
- Reasonable use of functions/methods to break large tasks up into smaller ones.
- Reasonable use of loops, decisions, etc.

*While each team member **must** meet the *basic* requirements of both a blue feature and a yellow feature, you may substitute some other feature from the "extra list" (features 9-17) for the *advanced* requirements.

Note: Instructor may decide to award an implementer of an exceptionally good feature more than 100% of the number of points shown above for that feature.