ITERATION 1
Submission and Assessment

Big Picture

- Learning objectives:
  - Understand and implement software requirements in the context of established project.
  - Use version control best practices during code development.
  - Adhere to a style guide.
  - Use unit tests to verify code.
  - Use doxygen to document code.
  - Use UML to document code.
  - Use text and images to document code.
  - Create self-documenting code.
  - Use sophisticated c++ concepts in implementation.

Assessment

Each iteration is worth

20% of your overall grade in this course.

- Big scale breakdown (out of 100%)
  - 10% UML draft
  - 10% Preliminary Code Submission (only functionality assessed)
  - 40% Documentation
  - 40% Implementation

Documentation : 40% of iteration 1 grade

- Design Document (for this iteration, equivalent to mainpage.h)
  - Prose (not lists), equivalent to about 1 page of text.
  - Audience: Other programmers not familiar with the project.
  - Describe the overall design. Make sure to point out the separation of graphics_arena_viewer and the arena. The robot is another important design element that should be described, including the important elements that it contains.
  - Save in mainpage.h in src

- UML of Your Iteration1 Design (save in docs)

- GOOD WRITING IS ...
  - Correct
  - Clear (without the reader having to look at code)
  - Well organized
  - Has good flow

Documentation : 40% of iteration 1 grade

- Style Compliance:
  - Assessed with cpplint
  - Spot check for naming
- Doxygen:
  - Doxygen comments in code.
  - Assessed by compiling into html. Visual inspection of document and code.
- Code Inspection for Documentation
  - Self-documenting with nomenclature and organization
  - Appropriate commenting
- Bug Report
  - Report requirements that were not attempted.
  - Report requirements that are not functional (comment out if causing compilation errors).
  - Report “buggy” behavior.
  - If you have a sense of why code is not functional or buggy, please report that – it will help TA’s give partial credit.

Implementation: 40% of iteration 1 grade

- Priority 1, 2, and 3 Requirements:
  - Priority 1 requirements weighted heavier than priority 2 and 3
  - Assessed with unit tests
  - Assessed with visual inspection of game playing
  - Assessed with visual inspection of code when requirement not functional
- Compilation and File Structure
  - Compiles on cselabs machine.
  - /project/iteration1/src and /project/iteration1/docs
  - NOT on github: build folder
  - Automated assessment.
- Github Commits
  - Best practice use of github (e.g. regular commits with reasonable messages)
  - Assessed using “history” on github.