Global Requirements

Fall 2018

Assignment Handins

These requirements must be met for every assignment you hand in, including the non-final weeks of multi-week projects. They represent our basic expectations for work done in this class, and if you do not meet them we will not give you credit for the assignment.

- Your handin directory must either be an Eclipse project or contain an ant build file that compiles your code to an executable Jar.
- You must hand in a README file explaining anything the TAs need to know about your code. At the minimum, it must contain:
  - A copy of the rubric for the assignment, which can be found at /course/cs1971/rubrics/<asgn>/grade.txt. This will be used for the following two items.
  - How to verify requirement. This will vary depending on the requirement, but can usually be satisfied by pointing to source files in your engine package or indicating how to find evidence while playing the game. Please give information for each requirement in the rubric. If the requirement isn't complete, just write "incomplete" as the verification.
  - Documentation of known bugs, or indication that there are none.
  - The approximate number of hours it took you to complete the assignment. This will help us gauge the difficulty of the assignments for future iterations of the class.
- You must hand in an INSTRUCTIONS file containing instructions for how to play your game.
- Your code must have no external libraries or other external dependencies other than the standard Java SDK (J2SE). This includes dependencies on other Eclipse projects! In other words, your projects must be completely self-contained. Exceptions may be granted on a case-by-case basis.
- Engine and game code must be separated into different packages.
- Engine and game code must be reasonably separate logically. It should be possible to create an entirely new and different game by reusing just your engine code.
- Engine code must be well-designed. Although we do not grade code style and comments, we do expect that you give careful thought to the design of your engine so that it can be reused from week to week without major refactoring. We will not give credit for an assignment that violates basic object-oriented design principles or seems excessively "hacky." Examples include making every field of your objects public or making your entire engine a single object.
- Game code must be reasonably designed. Since your game code will not be reused as often or for as long as your engine code, it is not as important to make it extensible and generic. Nevertheless, we still reserve the right to refuse credit for a game that is excessively badly designed, such as executing all of your game code inside a single method.

- Your code must work on department machines.
- It must take advantage of the screen space provided by larger window sizes – it cannot be drawn at a permanently fixed size.
- Your game must never go below 20 FPS.