

CS 494 Homework 1: Due 13 Feb 2006

For this homework, you will generate a proposal for a project that you will be working on throughout the semester. Essentially, you will be going through the inception phase for your project. You can work in groups of two or individually for this project. Working in groups is encouraged, however it will have more requirements and stricter grading guidelines in an effort to balance the workload with those working individually. Also, you will notice that the programming language to be used is not specified in this assignment – this is for two reasons. First, the language used should be independent of the choices made here. And second, I haven't decided on it yet.

Project idea

First, you need to come up with an idea for your project. This can be anything you want, as long as it follows the guidelines listed below. A few project examples are a swords-and-sorcery game, aka a “fantasy” game, or a simulation of a casino. These examples will be used below. Other ideas are a space trading game, network simulation, air space flight simulation, etc. Note that the system does not necessarily have to have user input (in the simulations, for example). Lastly, any of these rules are flexible if a convincing argument can be made why a rule should be bent (you need to speak to me before the assignment is due if you want to bend a rule).

Number of classes: The system should use many classes. There should be many objects of these various classes, and they should be interacting (communicating) with each other. The purpose of this requirement is so that the project has some serious OO programming. Thus, extensive GUIs or graphical displays are probably not appropriate (as a lot of the code tends to be managing those parts, and is often not OO). You are welcome to (and encouraged to) use GUIs and graphics – you just can't have most of your code be the GUI or graphical part. In the fantasy game, there would be classes for creatures, weapons, rooms, treasure, inventory, the player, etc. In the casino system, there would be classes for the cards, the workers, each individual game type (poker table, roulette wheel, etc.), etc. Note that similar classes (such as 1 deck blackjack and “normal” blackjack, or the various monsters) should be one class, with fields that differentiate the objects.

Project size: As a rough estimate of the size of the project, it should have 30-50 “real” classes. The “real” means that a tiny or trivial class (with very few methods or that is rather short overall) doesn't count. The final project should be from 5,000 – 10,000 lines long. The project should also be something that one can easily increase or decrease the scope of during the semester. This is to prevent you from being stuck with a project that is too big or too small. Group projects will need to be on the upper side of that range; individual projects on the lower side. For example, with the casino, the number of games can be increased or decreased; in the fantasy game, a number of less important aspects (inventory, map display, etc.) can be modified to achieve the same result. The idea of this homework is to do some planning of your project in advance. As we are not doing the waterfall method, we are not looking for exact requirements here. But you should think about all the classes and code length required so that you get an appropriately sized project.

Deliverables

You need to include the following in the document that you submit for this homework.

- **Vision:** A page or two describing the vision of the project – what does it do? How does it do it well? See the description (in the text and the slides) about what else should be in a vision statement. We are not looking for insincere platitudes about how great the system could be –

just an honest examination of the scope of this system.

- **Use cases:** you should list all the use cases you expect to use in this project. At least two of the use cases should be fully dressed (more, if you feel it's necessary for your project), the rest should be brief. You need to follow the use case guidelines in the textbook. In the casino game, each game would be a separate use case; entering data into a “bank” would be another, etc. For the fantasy game, each type of action (i.e. attack, managing inventory) would be a different use case. Not surprisingly, the fully dressed use cases will be worked on first, so make sure you choose the appropriate ones.
- **Supplementary specification:** You must include this as well. Details on the supplementary specification can be found in the textbook as well as what was discussed in lecture.
- **Actors:** A page describing the various actors in the system (which are probably going to be the interacting sub-systems of the project) and a UML diagram listing how the actors and the use cases interact (see slide 17 in the second set of slides for an example of this sort of diagram).
- **Miscellaneous:** Not a UP artifact, but important for this assignment. You need to discuss how you plan to increase or decrease the scope of the project if it becomes necessary (i.e. what specifically is going to be added or removed), and your language(s) of choice for implementing the system.

15 pages is a sample length for this assignment – 8 pages for the fully dressed use cases, 2 pages for all the other use cases (which are all brief), 2 pages for the actors, 2 pages for the supplementary specification, and 1 page for the miscellaneous section. While this may seem like a lot, remember that the fully dressed use cases (which are about half of your deliverables) are short-hand notation, and are not grammatically correct English (not that you shouldn't use bad grammar, mind you; you just don't have to write an essay here). Note that the lengths will vary from project to project. If your fully dressed use cases happen to be on the short side, include more of them. If your assignment is only 10 pages, then that's probably too short; 20 pages is probably too long.

You will notice that a lot of the finer details have not been laid out. For example, what should go in the supplementary specification? These details are being left to you, as they will vary from project idea to project idea. We are looking for (and will be grading based on) the fact that you have put a lot of thought into this system. A supplementary specification that just lists obvious concepts will receive a poor grade.

Submission

Because of the grading system we are using, the file names are required to be exact (this restriction may be lifted later in the semester, but it requires a lot of coding, and it's not as easy as it sounds). You will need to let me know BEFOREHAND if your file size is greater than 5 Mb, as the system does not allow files larger than that. The submission is done online at <http://www.cs.virginia.edu/~cs494/submit.html>. Details about using this system will be forthcoming. Groups need only submit one file for both people (when you submit the assignment, you will specify all the group member(s)).

All diagrams need to be done in Microsoft Visio. We will provide you a licensed copy if you do not have one. All of the deliverables must be in a single file, named hw1.odt. This is an OpenOffice file format – if you do not have OpenOffice, you can download it for free at <http://www.openoffice.org>. If you would rather work in another word processor (such as Word), that's fine – you can import it into OpenOffice right before you submit it (although you should check the formatting before you submit it). You are welcome to embed anything you want in the file (Visio diagrams, pictures, graphics, tables, etc.) – it just needs to be one file (and see above if it's bigger than 5 Mb). Lastly, a few formatting requirements: standard font size (12 point), single (sic) spaced, standard margins (1.25 inch side inches, 1 inch top/bottom margins), and start a new page for each separate deliverable. Whew!