CSC 454/554 First Assignment

Please complete the following assignment before our first class on Saturday September 16th, or you will face a harsh brand of justice. Seriously, our little class is only meeting 6 times and will be extremely time-constrained, so your preparation will be important.

Reading

Please read these chapters of "Design Patterns Explained" by Shalloway & Trott:

- Scan the Preface, but of course
- Chapter 1 "The Object-oriented Paradigm"
- Chapter 2 "The UML"
- Chapter 5 "An Introduction to Design Patterns"
- Chapter 9 "The Strategy Pattern"

OO Design Problem

Please use your new UML skills (see the reading above) to describe your classes, their methods and attributes, at each step. We'll sit in a circle, hold hands, and discuss your design ideas in class. Well, maybe we won't sit in a circle.

Let's go… You've been hired by that hot new video game startup Plutonium For Christmas, Inc. (PFC or "Plutonium" as it's called by the kids) to design their newest game: DuckSim. I guess the idea is that there are ducks and you do things with them or something. We'll figure it out.

Well, of course you'll have ducks in your game (duh), and more specifically two kinds of ducks: Mallard and RedHead. You can expect to have to support more kinds of ducks in the very near future (like the next step, eh). All ducks can quack and swim, and you'll want to be able to display your ducks on the screen in some way though a Mallard will display differently than a RedHead.

1. Describe the duck-related classes you feel will be necessary for DuckSim… in UML, please. [Hint: I only have three classes. I just don't want you thinking that this is harder than it is. This is the first assignment and before any lectures, so chillax.]

The President of Plutonium wants your ducks to fly as well. He points out, emphatically, "I mean ducks fly, don't they!"

Ironically, in the same meeting PFC's VP of Marketing chimes in that your game should support rubber ducks and decoy ducks as well. "The kids will love these, and it ties into that Motorola deal I'm working on."

2. Please add the ability to fly to your ducks. Also, how would you add the new rubber duck and decoy duck requirements to your design?

You may have noticed that we've got a bunch of kinds of ducks here (4 so far, and lord knows how many when we're through), but their behaviors are actually quite different:

- Rubber ducks actually squeak rather than quack
- Decoy ducks can't fly or quack at all for that matter

It would be nice to encapsulate these behaviors in our design as they somewhat volatile.

3. Study the Strategy pattern you read about in chapter 9. Try to apply the strategy pattern to your design (if you haven't already… yikes!), assuming that quacking and flying are "algorithms" as the chapter describes.

4. As a last step, it should be pretty easy (if your "getting" all this) for you to create C++ or Java classes from your last UML diagram. Do this and compile your code. Use stubs (empty methods) to get your code to compile.

That's it. Bring your stuff to class, and we'll compare notes.
BTW:

- Feel free to contact me about class or whatever at wtkrieger@noctrl.edu
- Our class web site will be up soon. Try http://william.krieger.faculty.noctrl.edu/
- If you know anyone who may be interested in 454, please let them know and have them contact me. Sell, sell, sell…
- I expect that you'll have a design problem of this ilk to solve for each class. One of my primary goals for CSC 454/554 is for you to get more comfortable making OO design decisions by practice, practice, practice.

See you Saturday!
Thanks, Bill

Created: August 10, 2006