

CSC 454/554 Homework #4

Created: October 5, 2006

Please use the same drill as the previous homework: one, two, three:

1. Take your best shot, sketch some UML and notes. (2 points)
2. Read up and then try to apply the pattern I give you. Again, your output should be UML and some notes. (2 points)
3. Implement your pattern-based solution in Java or C++ and place it on the k: drive. Please build it for me, so that I don't have to. (1 point)

So, homework #4 is worth 7 points. Wait, that doesn't add up?!? Read on.

In grading, I'll be looking for completeness and correctness, and a sign that you have spent some time working and thinking on this problem. So, I'll be looking for your notes and thoughts and other artifacts in addition to your final UML diagrams.

OO Design Problem: Mazes, factories of them

I want to build a maze game or two. Please don't get too caught up in details, rather focus on how mazes get created.

Let's agree that the topmost class in our hierarchy is `MazeGame`. A `MazeGame` object has-a `Maze`. Each `Maze` is composed of any number of `Room` objects.

This is tiresome. Look at my crappy UML drawing in Figure 1 for the overall idea. Also look at the code in Figure 2 for an idea of how rooms (each has 4 sides) are built, one side at a time.

I guess that I'd add:

- Each `MazeComponent` has an `enter` method, but the result of trying to enter a `Wall` will be quite different than entering an open `Door`.
- `Room` objects are numbered. So you can access the `Room` objects in a `Maze` by number.
- You set and access any of the 4 sides of a `Room` using a `Direction` enum defining `North`, `South`, `East` and `West`. I presume you'd use the same enum when traipsing through the `Maze`.

So, the idea is to build a maze and then play some game or other by moving through the maze. Remember, we're focusing on building the maze for this homework.

Another focal point is that we'll be building different types of mazes. An "enchanted" maze might be one where things appear and disappear. A

“dangerous” maze might have things that explode. A “console” maze might be a text-based game, ala Rogue from back in the day.

So, enough already... my question is, how would you create this variety of mazes and their contents?

For this homework, your three steps/deliverables are:

1. Show me your UML and notes on a design for this problem before reading the pertinent patterns chapter.
2. Read Chapter 11 of DPE on the **Abstract Factory pattern**. Complete a new (improved?) UML class diagram, with your notes on why you think things are better.
3. Code up your classes in Java or C++.

Regarding the pattern du jour... I would expect that you'll want a Factory for each of the kinds of different mazes we talked about: enchanted, dangerous, and console. Hopefully, your factories can make mazes and rooms and doors and more!

OK, we have two more points to earn. Question: How many instances of any concrete factory should usually exist in your program?

If your answer is one, then please read Chapter 21 on the **Singleton pattern**.

And for two points:

- Show UML incorporating the Singleton pattern into your design
- Code it up.

My email is wtkrieger@noctrl.edu

good luck... yow, bill