CPSC 2720 – Assignment 3 (Spring 2016)

OVERVIEW

In this assignment, you will:

- Design a simple system.
- Implement various design patterns.
- Keep track of your progress using version control.
- Document your implementation using doxygen.

INSTRUCTIONS

1. Fork the repository at https://bitbucket.org/ulethsecourse/asn3spring20167. Make sure that your forked repository is public (not private!) or your assignment will not be graded.
2. Create a local clone of your repository.
3. Design a system that solves the problem given below.
4. Implement your designed system.

PROBLEM DESCRIPTION

Following years of ineffective leadership, the USA has declined into a post-apocalyptic wasteland ruled by corporations who compete with each other in tournaments using mechanized robots (a.k.a. "mechs"). Each mech is made of a torso, arms and legs, where an arm or leg can hold one type and size of weapon. Table 1 shows the different sizes of torsos that your company manufactures, and Table 2 shows the different weapons that can be put onto either a mech's arm or leg. Table 3 shows examples of mechs the company has already put into production.

Table 1: Mech Torsos

	ARMS	LEGS	ARMOUR
SMALL	2	2	15
MEDIUM	4	2	25
LARGE	6	4	35

Table 2: Mech Weapons

	SIZE	DAMAGE	RECHARGE TIME	UPGRADES
LASER	SMALL	2	3	1
	MEDIUM	4	6	2
	LARGE	8	12	3
ROCKETS	SMALL	3	5	1
	MEDIUM	6	10	2
	LARGE	12	20	3

Table 3: Current Mech Models

	TORSO	LASERS	ROCKETS	PROGRAMMING
LOKI	SMALL	2 MEDIUM (Arms) 2 SMALL (Legs)	0	Aggressive
THOR	MEDIUM	2 MEDIUM (Arms) 2 MEDIUM (Legs)	2 MEDIUM (Arms)	Round-Robin
ODIN	LARGE	2 LARGE (Arms) 2 SMALL (Arms)	2 MEDIUM (Arms) 4 SMALL (Legs)	Round-Robin

Each mech can be programmed with a different battle strategy. Currently, the company only has two strategies implemented, but will implement more in the future:

1. ***Aggressive***. Always choose an available weapon that will do the most damage.
2. ***Round-Robin***. Fire each weapon once before firing a weapon again.

You have been asked by your company to create a mech simulator to test the success rates of mechs with different configurations of torsos and weapons. In the simulator, a mech battle is turn-based, meaning that each mech takes turn attacking. On a turn two things happen:

1. Weapons recharge by one time unit.
2. The mech fires one of its weapons. Which weapon fires will depend on the how the mech is programmed.

The company plans to have upgrades for the weapons in the near future. As the upgrades have not been built by the company, the simulator is not required at this time to simulate their application. [1] However, this will be happening in the near future and it may be worth implementing if you have time. There are two types of upgrades planned: one that reduces the recharge time and one that increases the damage at the expense of a longer recharge time. Table 4 shows details about the proposed upgrades.

[1] In other words, you don't have to do this for the assignment unless you are looking for more of a challenge or want to try for some extra credit.

Table 4: Proposed upgrades.

	SIZE	DAMAGE	RECHARGE TIME
FREEZE	SMALL	-	-25%
	MEDIUM	-	-33%%
	LARGE	-	-50%
INFERNO	SMALL	+25%	+10%
	MEDIUM	+33%	+20%
	LARGE	+50%	+30%

PART 1 – DESIGN PATTERNS

Create a UML static structure (i.e. class) diagram using VioletUML that shows the design of your software. You are expected to use the following design patterns in your design.

Design Pattern	*Functionality*
Builder	Used to assemble a mech.
Strategy	Used to provide the programming for mech.
Flyweight & Factory Method	Used to provide the parts for a mech.
Template Method	Used for implementing a turn.
Decorator (optional)	Used to apply upgrades to a weapon.

PART 2 – IMPLEMENT DESIGN

Implement your design from Part 1. It is suggested that you proceed as follows:

1. Implement the Builder pattern to assemble a mech.
2. Implement the Template and Strategy patterns to program the robot.
3. Implement the Flyweight/Factory Method to reduce the memory load of the simulation.
4. (Optional) Implement the Decorator Pattern to apply an upgrade to a weapon.

GRADING

You will be graded based on your demonstrated understanding of the use of version control, good software engineering design practices, and design patterns. Examples of items the grader will be looking for include (but are not limited to):

- Design and implementation shows use of design patterns.
- Version control history shows an iterative progression in completing the assignment.
- Implementation shows understanding of software engineering design principles.
- Source code is appropriately documented using the principles discussed in class (e.g. all classes and methods) so that `doxygen` can extract the documentation.

SUBMISSION

Submit the URL of your forked repository to Moodle. The grader will clone your repository as of the deadline for grading. Make sure that ALL of your files appear in the online repository.