Concordia University
Department of Computer Science
and Software Engineering

Advanced program design with C++
COMP 345 --- Fall 2014

Individual assignment #3

Deadline:	Friday, November 21st, 2014
Evaluation:	5% of final mark
Late submission:	not accepted
Teams:	this is an individual assignment

Problem statement

This is an individual assignment. It is divided into three distinct parts. Each individual student is expected to select one of these parts as his/her assignment. Each part is about the development of a part of the topic presented as the team project. Even though it is about the development of a part of your team project, each assignment has to be developed independently of the others and is not to be presented as an integrated part of the team project. Each member of your team is free to choose to do any part, and is expected to follow a different design approach than the other team members that have selected the same assignment topic. Note that the following descriptions describe the baseline of the assignment, and are related to the project description (see the course web page for a full description of the team project).

Part 1: Tower targeting strategy pattern

Use a Strategy pattern to enable to change the targeting strategy used by the different kinds of Towers. As a tower operates, it should first find a target within its firing range, then fire on this target. If more than one target is available within range, a choice must be made. Targeting strategies include *nearest critter to the tower*, *nearest critter to the exit point*, *strongest critter*, *weakest critter*.

Part 2: Critter wave factory pattern

Use a Factory pattern to create the waves of critters whose characteristics/type varies according to the number of waves destroyed thus far in the game. The assumption is that the further we are in the game, the harder it is to destroy the next wave of critters.

Part 3: Tower improvement decorator pattern

Use a Decorator pattern to provide a mechanism to alter the behavior of Towers as they are upgraded during play. The different kinds of concrete decorators can for example make a tower go up a level, changing e.g. its next level's cost, refund rate, range and power. Other decorators can be defined to add special damage effects to the tower, e.g. *splash* (damage also applies to critters near the target critter), *burning* (damage to the target critter is re-applied for a few seconds after a hit), *freezing* (damage to the target critter also slows down the critter).

Assignment submission requirements and procedure

You are expected to submit a group of C++ files implementing a solution to one of the problems stated above (Part 1, 2 or 3). Your code must include a *driver* that allows the marker to compile and execute your code on a standard lab computer. The driver should use the pattern to manage game objects and somehow demonstrate that the code conforms to the above-mentioned specifications, as well as following the applicable tower defense game rules, and that the patterns are correctly implemented. The use of unit testing such as cppUnit is not mandatory but encouraged. Along with your submitted code, you have to explain your analysis and design. Briefly explain the game rules involved in the creation of you assignment, citing external sources for specific game rules. Briefly describe the design you adopted as a solution. The design description can be backed-up, for example, by doxygen-generated documentation, regular code comments, or a simple diagram. The focus of this course being the coding aspect of software development, you are discouraged to submit extensive documentation.

You have to submit your assignment before midnight on the due date using the ENCS Electronic Assignment Submission system under the category "programming assignment 3". Late assignments are not accepted. The file submitted must be a .zip file containing all your code. You are allowed to use any C++ programming environment as long as they are available in the labs. No matter what programming environment you are using, you are responsible to give proper compilation and usage instructions to the marker in a README file to be included in the zip file.

Evaluation Criteria

Solution:	
Clarity and correctness of statement of game rules involved:	5 pts
Compliance of solution with stated problem:	20 pts
Simplicity and appropriateness of the solution:	5 pts
Clarity of design description:	5 pts
Programming style:	
Code readability: naming conventions, clarity:	5 pts
Coding style: `.h` and `.cpp` files, use of comments:	5 pts
Relevance of driver and presented results:	5 pts
Total	**50 pts**