Concordia University

Department of Computer Science
and Software Engineering
Advanced program design with C++

COMP 345 --- F2019 (Section N)
Assignment #4

Deadline: November 30th, 2019 by 11:55PM

Type: this is a team assignment (4 members max.)

Evaluation: 8% of the final mark

Submission: Electronic Submission through your Moodle course homepage. Create an appropriate zip archive including the entire C++ *source* files and all related deliverables.

Problem statement

This is a team assignment. It is divided into two distinct parts. 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 is to be developed/presented/tested separately. The description of each part portrays what are the features that the part should implement, and what you should demonstrate. 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, as well as links to the details of the game rules to be implemented.

Design Requirements (for all the two parts)
All the code developed in assignment 4 must stay in the same files as specified in assignment #1 and #2, and 3.

Part 1: Cards Factory Pattern

Use the factory design pattern to generate objects of concrete game cards implemented in part 4 of assignment#1.
You must deliver a driver that demonstrates the `Cardsfactory` interface and concrete classes implementing each type of game cards. The Factory and concrete classes code must be implemented using a new `Cardsfactory.h/Gamecard.cpp` file duo.

Part 2: Game Tournament

A tournament starts with the user choosing game map, and 2 -4 players with different strategies. Once started, the tournament plays all along from selecting the first player to actual game play, without interaction. At the end of the tournament, a report of the results should be displayed, and who is the winner, example of result:

Payer #	Cards	Victory points	Coins
1	6	4	8
2	4	3	15
3	10	16	18

In order to minimize run completion time, each game should be set to up to 30 turns, if no winner it is declared, the game is declared draw and prompt the results of each player.

You must deliver a driver that demonstrates that (1) when the game starts, the user is given the choice between single game mode and tournament mode; (2) when the tournament mode is chosen, the user is asked to select (i)the game map; (ii) from 2 to 4 players which can play any of the implemented strategies from assignment#3, including new customized strategy if you thought about one. The results of the tournament are displayed as depicted above with the winner player. This must be implemented in the `GameEngine.cpp/GameEngine.h` file duo.

Assignment submission requirements and procedure

You are expected to submit a group of C++ files implementing a solution to all the problems stated above (Part 1,2). Your code must include a *driver* (i.e. a main function) for each part that allows the marker to observe the execution of each part during the lab demonstration. Each driver should simply create the components described above and demonstrate that they behave as mentioned above.

You have to submit your assignment before midnight on the due date using Moodle under *A#4_SubmissionBox*. 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 you can demonstrate your assignment in the labs.

Evaluation Criteria

Knowledge/correctness of game rules:	2 pts (indicator 4.1)
Compliance of solution with stated problem (see description above):	10 pts (indicator 4.4)
Modularity/simplicity/clarity of the solution:	2 pts (indicator 4.3)
Mastery of language/tools/libraries:	4 pts (indicator 5.1)
Code readability: naming conventions, clarity of code, use of comments:	2 pts (indicator 7.3)
	Total 20 pts (indicator 6.4)

Reference

[1] "Game rules." Red Raven, game Design: Ryan Laukat Illustration: Ryan Laukat, Alex Davis, John Breckenridge, Malorie Laukat ©2012 Red Raven Games.
https://redravengames.squarespace.com/eightminute-empire