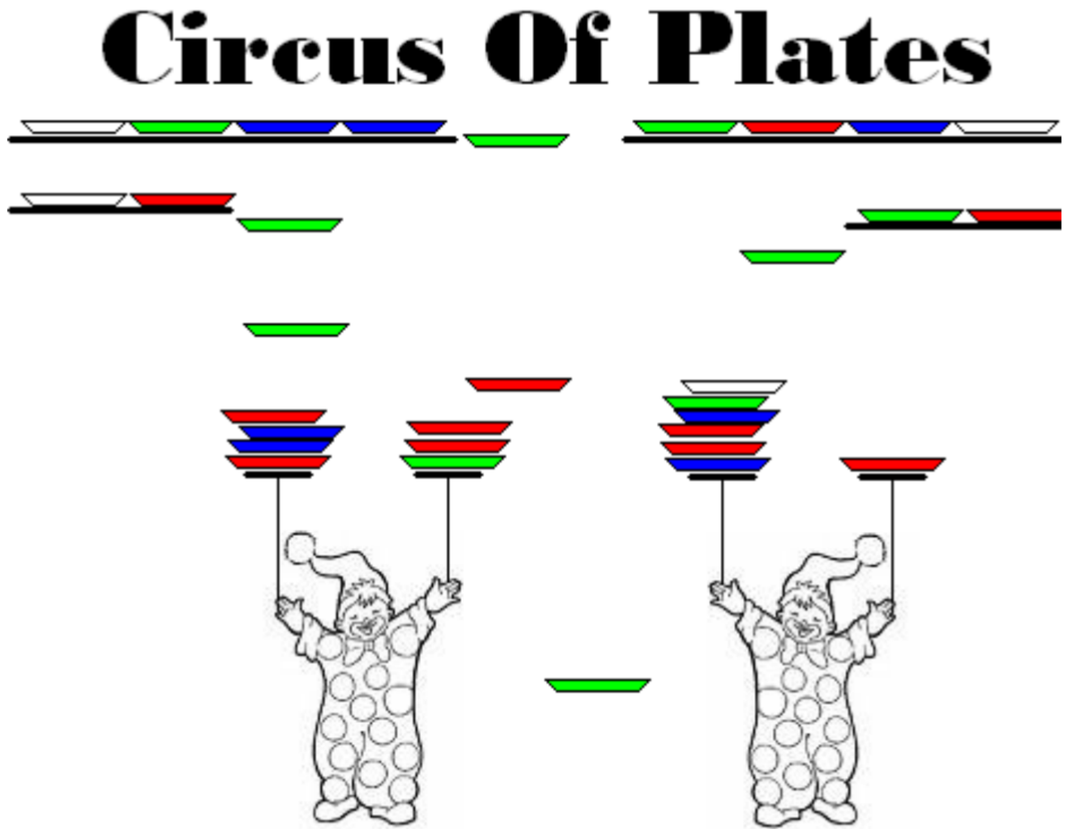
Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.
Second Year
Fall 2016

CS221: Object Oriented Programming
Programming Assignment 5
Due Date: Thursday, Jan.25th 11:59 pm.

Circus of Plates - Game

It is two player-game (one may use keyboard, and the other uses the mouse) in which each clown carry two stacks of plates, and there are a set of colored plates queues that end up falling down. Players try to catch the falling plates, if they manage to collect three consecutive plates of the same color, then they are vanished and their score increases. You are free to put rules ending the game.



Tasks

- You should not support only plates; you should support other shapes (you should have a class Shape). The shapes classes should be dynamically loaded at the start of the execution from a specific folder. You should support at least two shapes.

- The user gets a point when he collects three consecutive shapes from the same color (even if they are different shapes).

- You should use (at least) the following ten patterns in your design:
  1. Singleton,
  2. Factory or Pool,
  3. Iterator,
  4. Dynamic Linkage,
  5. Snapshot,
  6. State,
  7. Strategy,
  8. Observer,
  9. MVC
  10. Object Pool
  11. and choose another one you feel it suits your design.

  This assignment mainly tackles the application of what you studied in the course of design patterns, so you are supposed to give time for the design.

  Using MVC means you should isolate the three main parts of the design. Model, View and Controller.

- Complete log of the operations done on the game (e.g., user actions, objects intersections, creation or reusing of objects) should be generated. You must use log4J package for this purpose. The Log4J Open Source package could be used for that purpose. Logging all such actions will introduce a delay to the game, so you must use multiple level of logging (e.g. DEBUG, INFO, ...), and classify your logs according its nature.

- You need to support at least 3 levels of difficulties, but you are free to choose any criteria for difficulty (e.g., different speed, multiple clowns, more queues, changing plates colors or sizes, ... etc.).

- You are free (In fact, encouraged) to spice the logic of the game as much as you want, add features or scenarios that are not required here, provided that you implemented the required features.
- This assignment will NOT be tested by the OnlineTester.
- You should support saving and loading the game, in any format you prefer.
- You should support pausing and resuming the game.

Report

The report should contain the following:

- Describe your design thoroughly.
- Class diagram of your design.
- Sequence diagram showing the typical scenarios of the game.
- Section for each pattern (the required and any other patterns you used) and how you used it in your design, and a class diagram explaining this.
- Snapshots of your GUI.
- User guide explains how to play the game.
- Any design decisions that you have made should be listed clearly.
- A significant part of the grade will go to the report, so make sure you make it clean and readable.

Notes

- You should work in groups of up to **four.**
- You should pack your game in an executable JAR file.
- This assignment mainly tackles the design issues. Heavy load will be on the good design in addition to the required patterns and good report.
- Develop this assignment in Java.
- You should deliver your source code using your git repository.
- Delivering a copy will be severely penalized for both parties, so delivering nothing is much better than delivering a copy.