WESTERN UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SE2250b – Software Construction

Assignment 3: Prospector Solitaire

Deadlines:

Section 002: Tuesday, March 19th, 2018
Section 003: Monday, March 18th, 2018

NOTE: Proper coding practices should be used. Examples include, but are not limited to: following naming conventions, project organization, proper code formatting, clean code (no old commented out code or unnecessary code), and any other topics discussed in lectures or labs.

In this assignment you will be working with an existing application. The application is not working properly, and it is your task to identify problems and fix them. You will also explore Unity limitations and integration with other systems/software. To start with, download "Prospector Solitaire-Broken" project from OWL.

1. (8 points) Five mistakes have been integrated into a working solution of Prospector Solitaire. You need to identify, understand, and fix these problems. This does not mean that the code needs to be changed in only five places. Each mistake may have repercussions elsewhere. You cannot modify the code within CardProspector.

   In order to receive the full mark, the changes must follow proper software design (inheritance, polymorphism, encapsulation, patterns and similar).

   Note 1: Clicking on the draw pile (the pile in the right, upper corner) must result in taking the next card from the draw pile and putting it on the target pile (the pile in the middle where the cards are ordered). Clicking on the target pile should not do anything.

   Note 1: The game must start with random order of cards in the deck.

   1.a) (4 points) Identify the problems and explain why the code did not work as required.
   1.b) (4 points) Fix these problems.

2. (4 points) Now that you have a working version of the code. The following modifications must be made.

   2.a) (3 points) You must ensure that the cards are of alternating color for a move to be accepted.
   2.b) (1 point) You must print to console the cards name (H10 for 10 of Hearts is sufficient) of the target as it changes, however you cannot modify the code in the CardProspector script.

3. (4 points) Integrate Unity with one of the social media platforms (Facebook, Twitter, or any other one) by providing the ability to share your Prospector Solitaire game score on the chosen social media platform. Typically, you would share a high score, but make sure that you can share even if not high score so you can demonstrate in the lab. Another option is to share screenshots from your game. The process can be integrated with a web browser.
4. (4 points) Identify at least three limitations of Unity. These can be things you cannot do in Unity itself but need to use external resources. Explain each identified limitation as well as a possible way(s) to overcome it.

Tips:

- You can assume the scoring scheme is correct.
- There is a pre-set high score, that is fine.

Deliverables:

- Functionality demonstrated to a TA during the labs.
- All game files uploaded to GIT and to OWL. This should contain all files needed to run your game from UNITY.
- A PDF file uploaded to OWL containing an answer to question 1.a (make sure your name is included in the file header)
- In OWL submission (text field) provide a reference to Git submission.
- Your GIT repository name and structure must follow instructions from the Git tutorial (Intro_to_Git_2019.pdf).

If the student did not demo the solution to TA during the lab section on the due date or before, the student must contact the professor within the two days following the deadline to arrange the time for the demo.