CSCI 2300, Fall 2019

Team Project

Phase 2: User Interfaces and Design [100 points]

Overview

For this part of the team project you need to design your application. After completing this phase, you will have a high-level implementation of the various use cases that your application is required to support, a sketch of your class interfaces, and an understanding of how different classes of your application fit together.

In class, we created a design for the Sliding Puzzle application. I recommend you follow the same process as we did in class to create the design of your team's application.

Details

Break down your requirements into specific use cases: possible user actions. Write down specifics of what is expected in each use cases. Determine what classes you need to create and what methods these classes need, based on your use cases and requirements. Code up the stubs for the needed methods of your classes (return type, method name, parameters) and include a "dummy" implementation (the methods do not yet have to be fully functional, just enough code to make it compile).

Provide high level implementation for each use case: make calls to the appropriate methods of different classes. Since not all class methods are fully functional yet, it is expected that your use cases will not yet be fully functional.

Review your classes and use case implementation and create a class diagram for your design. Some IDEs provide tools that will generate class diagrams from your code. Another alternative is to create class diagrams manually. The tool I used for example diagrams distributed in class is <u>https://www.draw.io/</u>.

<u>Submit</u>

- 1. A pdf document with an English description of your use cases and the expected behavior of each use case (via email, one per team).
- 2. Class interfaces and high level implementation of the use cases (via git, to your team git repo: create model and ui subdirectories and place your classes in the appropriate subdirectory; if necessary, you can create additional subdirectories.). Your code needs to compile but does not need to be fully functional yet.
- 3. A pdf of your class diagram (via email, one per team).
- 4. Peer-evaluation for your team members (each team member must submit a peer-evaluation form to receive a grade for this phase of the project; via email or a hard copy in class).

Grading

Your grade will be based on the following criteria:

[20 points] Use cases pdf document: do your use cases match requirements?

[40 points] Use case implementation and class interfaces: does your high-level implementation of use cases cover all the required behavior? Do your supporting classes have the necessary interfaces?

[20 points] Class Diagram: does your class diagram match your code?

[20 points] Peer evaluation results: a portion of your grade will be determined by how well you worked with the rest of the team and your contribution to the project (as evaluated by your teammates).