Milestone: Design

OVERVIEW

During this milestone, you will:

- Create a design for a software system using:
  - Model-View-Controller
  - Dependency Injection
- Collaborate with other teams to develop the final design.

SPRINTS

The milestone will be completed in two 1-week sprints. Sprint activities are completed as a team.

Sprint 1

1. Examine the user stories found as issues in the Master repository for the game.
2. Perform OOAD to determine the classes and their respective methods and attributes for implementing the game.
   a. The design is to follow the SOLID+DRY principles. Refer to the Class Notebook to refresh your memory of these principles.
   b. Identify the classes that will have the roles of `Model`, `View`, and `Controller` within your application.
   c. The Controller is to use Dependency Injection for its `Model` and `View` dependencies.
      i. Choose between constructor injection, setter injection, or a combiniation of them both.
3. Create a UML class diagram for the system.
   a. Identify the roles of the class using the stereotype annotation.
4. Create a sequence diagram for each user story or set of user stories that logically go together.
   a. If a set of user stories would have the same sequence diagram (e.g. dealing cards to 2-3 players and dealing cards to 4-5 players), only one diagram needs to be provided.
   b. Expected sequence diagrams include, but are not limited to:
      i. Dealing cards
      ii. A player's turn
      iii. End of game
5. Commit the following files to a `design` directory in your team's repository:
   a. `Class Diagram.pdf` – The OO design for the game.
   b. `<User Story Summary>.pdf` – A sequence diagram for one of the user stories (e.g. `Deal Cards.pdf, Player Turn.pdf`)
      i. The project is to contain a file for each sequence diagram.

Grading

The sprint will be graded as Completed/Mostly Completed/Not Completed.

Sprint 2

1. Examine the class and sequence diagrams of the other two teams.
2. Determine how the team's design could be improved. Examples include:
    a. Better use of the MVC architectural pattern.
    b. Better implementation of DI.
3. Revise the team's design documents to reflect the identified changes.

4. Select one team member to be a Scrum Master to meet with the other Scrum Masters for the same game.
5. The Scrum Masters examine the team's revised designs to create a `Class Diagram.pdf` and appropriate sequence diagram files in the `Master` repository on the `design` branch for the game.
    a. Other team members may attend the meeting but should be "Chickens," not "Pigs."
    b. Scrum Masters may want to divide the work of creating the files between the different teams.
    c. At the end of the sprint, the instructor will use this file to transfer the listed issues into the game's Master repository for the Design milestone.
6. Create a pull request in the game's `Master` repository from the `design` branch to the `main` branch.

Sprint Grading

The sprint will be graded on the following criteria:

a. Individual Teams:
    a. Class Diagram [20 marks]
        i. Classes are annotated by their role in the MVC architecture pattern.
        ii. Controller classes show the use of DI.
    b. Sequence Diagrams [20 marks]
        i. Each user story has a corresponding sequence diagram.
            1. Multiple user stories may correspond to the same diagram.
        ii. Collectively, the sequence diagrams cover all
b. Game Teams (all teams for a game share this grade) [20 marks]:
    a. Completed class diagram
    b. Completed sequence diagrams

MILESTONE DELIVERABLES

Sprint 1	Class and sequence diagrams
Sprint 2	Updated Class and sequence diagrams (each team) Final class and sequence diagrams for the game