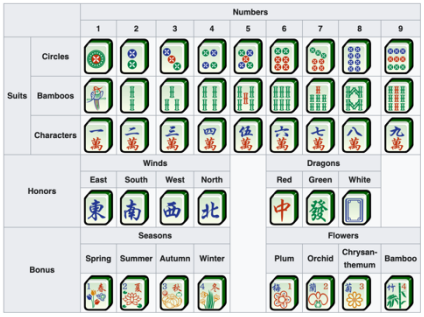
Software Engineering Project - Mahjong tiles Game(Beijing Version)



Students required to elicit the requirements of the Mahjong tiles game (Beijing version) and implement the game using Java following the software engineering guidelines.

Evaluation Criteria:

The project will be evaluated along the following criteria:

Phase-1

- **Game requirements and Modelling (20%)**
  - User requirements and system requirements of the Game are elicited correctly.
  - The Functional and non-functional requirements identified correctly.
  - It would be useful to show that you have created use case diagrams, sequence diagrams and class diagrams of your project (for drawing the UML Diagrams: I would recommend you to use **Papyrus since it will be covered in the lab session.** However, feel free to use other tools like https://sequencediagram.org/ and https://app.genmymodel.com/).

Phase-2

- **Clean design (20%)**
  - Game components are implemented appropriately, i.e. separate classes are defined to create different game components.
  - Is the project well-structured in classes and modules? Each class should perform a small set of related functionalities (No big classes, long methods)
  - Are interfaces defined and used appropriately?(Interfaces must be created for classes showing similar behaviour but not extending the same superclass)
  - Use the correct data structures
  - Adopt inheritance wisely (e.g., to distinguish between different types of players).

- Design patterns (e.g., Singleton, Facade) must be used when necessary. In this assignment, it may be necessary to use the following design patterns: Singleton, Facade, Observer, and Model-View-Controller.

    - **You have to use at least five design pattern types!**

- **Functionality (40%)**

    - The functional requirements of the game are implemented correctly.

    - Consider having GUI for the game.

- **Teamwork (10%)**

    - The distribution of load between teammates is appropriate

    - Distributed version repository (Git) is used appropriately

    - Try to avoid situations where one team member did all the work and another just went for the ride. These things can be understood very easily during the correction of your project.

    - Make sure that your commit messages are self-explanatory and describe the type of modification that was performed in the project.

    - Commits should be performed by both team members frequently.

- **Testing (10%)**

    - Several meaningful test cases are implemented to test the main project functionalities.

    - Ensure that the main methods of each class are tested: no need to test getters and setters; ensure that essential methods (e.g., those you decided to show in the sequence diagram are tested).

Project Registration (by Week 2):

- One group project (4 people)
- You can choose who to work with
- Each team member will receive the same evaluation (although there may be exceptions.)
- If you did not choose your group member, then I will create new groups and sign you into these groups randomly!

https://docs.google.com/spreadsheets/d/1C2M-_3AdmtjX1OMlZPwY0PGf5zmm-ecx1ggftgcKDNQ/edit#gid=0

Important Dates:

The deadline is on the **end of week 15.** You need to do the following for the submission:

1. **Phase-1 (Deadline: Week 10 ,on Monday 04/29/2024)**

   Provide a pdf file contain the following elements:

   - Game requirements.
   - Provide the UML design of your project.
   - Project completion plan (timeline to finish the project by Week 15).
   - Provide the member contributions and how you divide the work.

2. **Phase-2 (Deadline: Week 15, on Friday 06/07/2024)**

   Provide a pdf file contain the following elements:

   - Provide any explanation you want to help me understand how you conduct the project considering the Game requirements and the concepts we discussed in the class about the best practice of being a software engineer?
   - Provide the member contributions and how you divide the work.
   - **Provide a link to your GitHub repository on Brightspace:** When submitting your assignment on Brightspace, you only need to provide a link to your Github repository in the pdf. Please, do not attempt to upload your code on Brightspace.
   - Please add me and the head of the TA (Alkabashi) as a collaborator to your Github project. Alkabashi will explain more about this in the lab session.

3. **Interview**: You will present your work (Demo) and answer some questions form the TA and me. **More details about the timetable for the interview will be posted due to the course.**

GOOD LUCK!