Assessment criteria

No	Criteria	Description	Score	Comments
1.	Completeness of the solution direction (Functional completeness and functional correctness)	to what extent are all key functionalities covered in the design, key functionalities covered are able to work perfectly	1-5	-
2.	Rationale behind the chosen solution direction (Functional appropriateness)	Justifications of choices made	1-5	-
3.	Understandability of the solution direction (Appropriateness recognizability)	Documentation quality and code readability	1-5	-
4.	Extensibility of the solution direction (Modifiability)	Design modularity and scalability	1-5	-
5.	Code Quality (Maintainability)	Quality of the code written	1-5	-
6.	User interface aesthetics (User engagement)	Aesthetics of UI	1-5	-

Prototype review for Yee Perng Yew

No	Criteria	Description	Score (1-5)	Comments
1.	Completeness of the solution direction	to what extent are all key functionalities covered in the design	3	Flip card and initial board implemented, yet to implement the rest of the rest of the key functionalities
2.	Rationale behind the chosen solution direction	Justifications of choices made	4	Solution approach is decently done in a class by storing data of the image in the class and getting the data of the image to perform a flip function.
3.	Understandability of the solution direction	Documentation quality and code readability	3	Minimal in-line documentation, code is easy to understand and user friendly.
4.	Extensibility of the solution direction	Design modularity and scalability	4	Modular design, though some areas could be more decoupled.
5.	Code Quality	Quality of the code written	4	lacks some of the OOP principles but takes into account of code reusability and simplicity
6.	User interface aesthetics	Aesthetics of UI	3	Simple and user friendly UI, however does not have a main menu page.

Prototype review for Ming Wei Yeoh

No	Criteria	Description	Score (1-5)	Comments
1.	Completeness of the solution direction	to what extent are all key functionalities covered in the design	3	Flip card and initial board implemented, yet to implement the rest of the rest of the key functionalities
2.	Rationale behind the chosen solution direction	Justifications of choices made	3	Solution approach is decently done, approach is done by using a popup to display the flipped card.
3.	Understandability of the solution direction	Documentation quality and code readability	2	No in line documentation, however code is self-explanatory.
4.	Extensibility of the solution direction	Design modularity and scalability	4	Modular design, though some areas could be more decoupled.
5.	Code Quality	Quality of the code written	5	Adheres to most of the OOP principles, well written code with simplicity and reusability
6.	User interface aesthetics	Aesthetics of UI	5	Simple and user friendly UI with main menu and player selection scenes.

Prototype review for Eng Jie Lee

No	Criteria	Description	Score (1-5)	Comments
1.	Completeness of the solution direction	to what extent are all key functionalities covered in the design	4	Flip card, initial board and player turn implemented, yet to implement the rest of the rest of the key functionalities
2.	Rationale behind the chosen solution direction	Justifications of choices made	4	Solution approach is well done by using colours to cover the content of the chit card and uncovers when the card is clicked.
3.	Understandability of the solution direction	Documentation quality and code readability	3	No in line documentation, however code is understandable with good naming conventions.
4.	Extensibility of the solution direction	Design modularity and scalability	4	Modular design, though some areas could be more decoupled.
5.	Code Quality	Quality of the code written	3	Adheres to some of the OOP principles, but lacking some of the principles. Has code reusability features.
6.	User interface aesthetics	Aesthetics of UI	3	Simple and user friendly UI with main menu and player selection scenes.

Prototype review for Avril Yi Qian Chai

No	Criteria	Description	Score (1-5)	Comments
1.	Completeness of the solution direction	to what extent are all key functionalities covered in the design	3	Flip card and initial board implemented, partial implementation of player movement, yet to implement the rest of the rest of the key functionalities
2.	Rationale behind the chosen solution direction	Justifications of choices made	4	Solution approach is decently done, by storing images as an attribute in a class and linking it with its respective image to perform a flip action when clicked.
3.	Understandability of the solution direction	Documentation quality and code readability	4	Some in line documentation, however code uses good naming convention and code is readable.
4.	Extensibility of the solution direction	Design modularity and scalability	3	Modular design, though some areas could be more improved.
5.	Code Quality	Quality of the code written	4	Adheres to a substantial amount of the OOP principles, decent written code shows simplicity and reusability, but repetition exists in some areas.
6.	User interface aesthetics	Aesthetics of UI	5	Simple colour coded, user friendly and easily understandable UI.

Overall Summary

Yee Perng Yew: Good design with user friendly UI but lacks some OOP principles and has minimal code documentation.

Ming Wei Yeoh: Excellent code quality and UI aesthetics but needs better in-line documentation and OOP principles.

Eng Jie Lee: Comprehensive functionality with a well-documented approach but requires improvement in code quality and UI engagement.

Avril Yi Qian Chai: Good UI aesthetics with readable code, but needs to improve modular design.

Metrics

Score for criteria 1

- 5: All key functionalities are fully implemented.
- 4: Most key functionalities are implemented, with minor omissions.
- 3: Some key functionalities are implemented, but significant omissions exist.
- 2: Few key functionalities are implemented.
- 1: No key functionalities are implemented.

Score for criteria 2

- 5: Solution approach is highly appropriate and efficient.
- 4: Solution approach is mostly appropriate, with minor inefficiencies.
- 3: Solution approach is somewhat appropriate, with noticeable inefficiencies.
- 2: Solution approach is minimally appropriate.
- 1: Solution approach is inappropriate.

Score for criteria 3

- 5: Documentation is clear, comprehensive, and easy to understand.
- 4: Documentation is mostly clear and comprehensive, with minor gaps.
- 3: Documentation is somewhat clear but has significant gaps.
- 2: Documentation is unclear and incomplete.
- 1: Documentation is missing or unusable.

Score for criteria 4

- 5: Design is highly modular, facilitating easy modifications.
- 4: Design is mostly modular, with minor difficulties in modifications.
- 3: Design is somewhat modular, with noticeable difficulties in modifications.
- 2: Design is minimally modular.
- 1: Design is not modular.

Score for criteria 5

- 5: Code adheres to standards, is simple and readable, with minimal case analysis/down-casts.
- 4: Code mostly adheres to standards, is fairly simple and readable, with minor use of case analysis/down-casts.
- 3: Code somewhat adheres to standards, with noticeable complexity and case analysis/down-casts.
- 2: Code minimally adheres to standards, is complex and hard to read.
- 1: Code does not adhere to standards, is very complex, and relies heavily on case analysis/down-casts.

Score for criteria 6

- 5: UI is highly engaging and visually appealing, with excellent UX.
- 4: UI is mostly engaging and visually appealing, with good UX.
- 3: UI is somewhat engaging and visually appealing, with acceptable UX.
- 2: UI is minimally engaging and visually appealing, with poor UX.
- 1: UI is not engaging or visually appealing, with very poor UX

Outline on creation of Sprint 3

Based on the outcome of the assessment, we decided to use Ming Wei's Code as our base code for sprint 3 since he scored the most based on the assessment criteria we conducted. However, we integrated our ideas into his code so that the code consists of ideas and elements from each of the members from the team. For the initial board, we took Ming Wei's idea as his board was perfectly done by calculating the X and Y coordinates by making use of the in-built library in JavaFX, geometry. For flipping cards, we decided to use Yee Perng's idea since it was neatly done by using images representing the number and types of dragons. As for player turn, the team decided to use Eng Jie's idea as it fits well while integrating with our code. Following player movement, we decided to use Avril's idea, a partial implementation done from sprint 2. Although it was partially done, we believed that it can easily fit the current code after slight modifications were made. When combining the ideas to one, modifications were made so that the different code blocks fit and work perfectly when combined.

We also got some new ideas while working on our sprint 3. After thorough discussion within the team, we decided to do things based on simplicity. Therefore, we decided to change the images to texts for the chit cards we created during sprint 2 for sprint 3. We also added superclasses such as the Dragon class to adhere to inheritance in OOP principles. This improves our code efficiency and reusability if needed in the future.