PEER REVIEW

Project Name:	Prova finale di Ingegneria del Software
Reviewed Project Members:	Simone Cervini, Ludovica Cova, Davide Fugazza
Review Author:	Aleksandro Aliaj, Leonardo Cesani, Matteo Colella
Review Date:	04/02/2022

Strong Points

Title	Annotations	Suggestions
Diagram's clarity	The diagram is well divided by abstraction	Align horizontally the characters in the decorator pattern
Interfaces	Good interface usage	None
Decorator pattern	Great pattern choice	None
Characters	Smart characters division	None

Weak Points

Title	Annotations	Suggestions
Association	The associations are too implicit	It would be better to specify the association's type between classes.
Student & professor	Few attributes and no methods	With more attributes you can specify more characteristics. For example, it is difficult to find the position of each piece.
No towers and coins classes	Easy implementation, but not Object Oriented	It would be difficult to manage the towers' movements without a class.
Game initialization	Implicit game initialization	It would be useful to add a class which manages the creation of the board (islands, clouds, …)

Cards & Decks	No deck managing	Introducing deck classes could be useful in managing the card's creation and selection
Int usage	Many attributes are managed with int.	Code would be more readable and manageable using different types instead of int values

Architecture comparison

- The most significant difference we noticed between our diagram and the one reviewed is the smart usage of interfaces to manage different method implementation, so this, for us is the most important feature we should take note to improve our architecture as well.
- Another important feature we noticed that is significantly different from our design is the division between simple and complex characters. We could take note of this idea to improve our code removing many constructors and methods which we used for every different character.
- We also noticed that the two architectures present a strong difference in the management of small classes, we followed a more object-oriented view, associating a class to every item of the game, on the other hand the architecture we reviewed does not share our idea. In fact, smaller items, for example coins, are implemented only as an int attribute in one class. We think that our idea is more coherent with the language used, but we can also take inspiration to make our code easier to implement.

Conclusion

The UML diagram we received is very clear and well organized; the weak point we found are not severe problems, but considerations based on the differences with our project. Overall, this review has been useful because comparing this project to ours has given us some ideas to improve our work.