Peer-Review 1: UML

Andrea Sgobbi, Roberto Scardia, Jonatan Sciaky, Luca Simei
Group 5

1 aprile 2023

Valutazione del diagramma UML delle classi del gruppo 45.

1 Positive Aspects

- The implementation of an abstract class ***CommonGoals*** is a great idea due to the fact that it will be easier to extend in the future.
- The enum ***GameState*** is useful when the model needs to be serialized. We recommend maintaining these states but to update them with a method in the Controller.
- Implementing a JSON file for the ***PersonalGoal*** class opens the model up to easier serialization in the future. This improves readability and simplifies testing.
- The use of ***ItemMatrix*** to unify Personal Goals and Shelves is very principled. It would work even better if this was also used by the board.

2 Negative Aspects

- Structuring the ***Item*** class with a single enumeration with three values for each type of Item complicates the evaluation logic of personal and common goals. We recommend creating an entity that separates the type of the Item from the information of the sprite. Furthermore, we don't understand why there is another enumeration for the color of the tile, given that each kind of item has a predefined coloration.

- Making the ***voidItem*** a subclass that only fixes the value of the color field is unnecessary.
- The ***Coord*** type is extensively used in the UML diagram but not defined anywhere.
- In the ***Board*** class it is not clear how the logic for legal coordinates is structured, since a matrix is being used (by this we mean coordinates banned based on the number of players).
- Since the ***Scorer***'s function is only applying the game's logic, and it does not actually represent any of the game's state, we recommend either moving it over to the controller or implementing each method in the respective class.
- In the ***Game*** class, there are functions (such as changeState and updateScore) that are changing the state of the model, thus they should be located in the Controller, not in the Model.
- The ***Color*** enum and the ***Type*** enum are redundant. We suggest to implement an enumeration for each different types of tiles, and to remember that any kind of tile can have three different sprites.

3 Architecture comparison

We found some good ideas in the CommonGoals and PersonalGoal classes. Using an abstract class to implement all the 12 diffferent common goals is a great way to extend them in the future, and very similar to the way we also implement them. We agree with the use of JSON for the goals, although instead of using it for initialization we use it to bundle tests.

Overall, we feel that the model is able to fully satisfy the game's requirements, although we think that some of the logic should be extracted and moved to the controller to adhere more strictly to the MVC pattern.