

Peer-Review 1: UML

Andrea Rovelli, Alessandro Pessina, Francesco Riccardi, Luca Tosetti

Group 03

Evaluation of the UML classes diagram of group 43.

Positives

A positive note is given by the detailed breakdown of the various classes, which allows a more precise control of what you want to do. Considering a perspective of future implementation of new functions, this allows to have extensible and integrable modules with each other.

Negatives

The UML diagram does not show a way to understand which common objectives have already been completed by the players: there is a group of methods to update the score, but if a player has already completed a common goal, he may be assigned another score tile for the same common goal.

In the Game class there is the ended attribute, but it is redundant because the combination at the end of the turn of `EndGameToken.isTaken()` and `active player == first player` is enough to obtain the same data. It is also unclear how it is determined who is the active player in a given turn.

There is no attribute in the ItemTile class that identifies the image associated to a single tile, which causes individual players to get different views.

In the Player class the completion of the personal goal is checked, but having the PersonalGoalCard class available it would make more sense to implement it within it.

In the ShelfSlot (or BoardSlot) class there is the empty (or Full) attribute to indicate whether the cell is empty or not, but it would be enough to manage the null value within the Tile (or Item) attribute, or check for the absence of assigned type through `Tile.getTileType()` (or `Tile.getTileType()`), following the implementation.

Comparing architectures

An interesting choice is to use an attribute to figure out if a cell is usable or not, which results in a clearer and cleaner structure than our combination of null cell and available cell, but without color.