Peer-Review 1: UML

Galparoli, Lucci, Milazzo

Group 55

Group 54 UML Class Diagram Evaluation

# Positive Things:

Show in this section what are, in your opinion, the positive things on the other group UML diagram.

UML Diagram, both in his Model and Controller parts, stick perfectly to the MVC pattern:

-Model is composed by all and only classes that contains game data and has methods that access them.

-Controller implements the game logic and possess methods that manage all the phases of the match (Game Setup, Round progress and Endgame).

In detail, the Model part is easily understandable and all the classes are well linked one to the other, therefore the game model is clear.

Last, explanation for the round progress, with reference to all the methods used in every phase, is really helpful and make easy to comprehend the order of invocation of the main methods and their effects on the match.

# Negative Things:

As in the previous section, show what you think are the negative things.

In the Controller Part, classes are not linked, making difficult to understand how they interact during the process.

In the Model, the creation of the “School Player Board” divided in three different classes, one for every part (Entrance, School Hall and Professor’s Hall) could be managed by putting them together in one collective class.

# Architecture Confrontation:

Identify Key Points in the Architecture of the other group compared to yours, and what are the adjustments that you can make to your Architecture to improve it.

In the Model part, both UML diagrams are pretty similar, with minor differences in the way Game Turns are managed; in particular, *round, playerOrder* and *currentPlayer* attributes that are inside the *Game* class make easy the Round Managment, so we will add them to our *Game* class, in association with their respective methods, to improve our Round Order Control.

Another Key Point that we could take inspiration by to improve our Diagram is the existence of the “Process” section, that make it easy the General Understanding of the Method invocation in the Program.