

Steps/Process

In the process in developing the simulation I have analyzed the requirements from the case description. Based on these requirements I have used the noun and verb analysis in some extent to identify potential classes and methods for these classes. Furthermore, I have implemented these requirements based on a temporary UML class diagram. I have afterwards tested one of the requirements implemented and gone further to the next. If there was a problem while testing, there would be a revision of the requirements and afterwards remodel them based on the new revision. So the process has been iterative.

UML Diagram and Rationale for the design

The UML diagram constructed based on the simulation application:

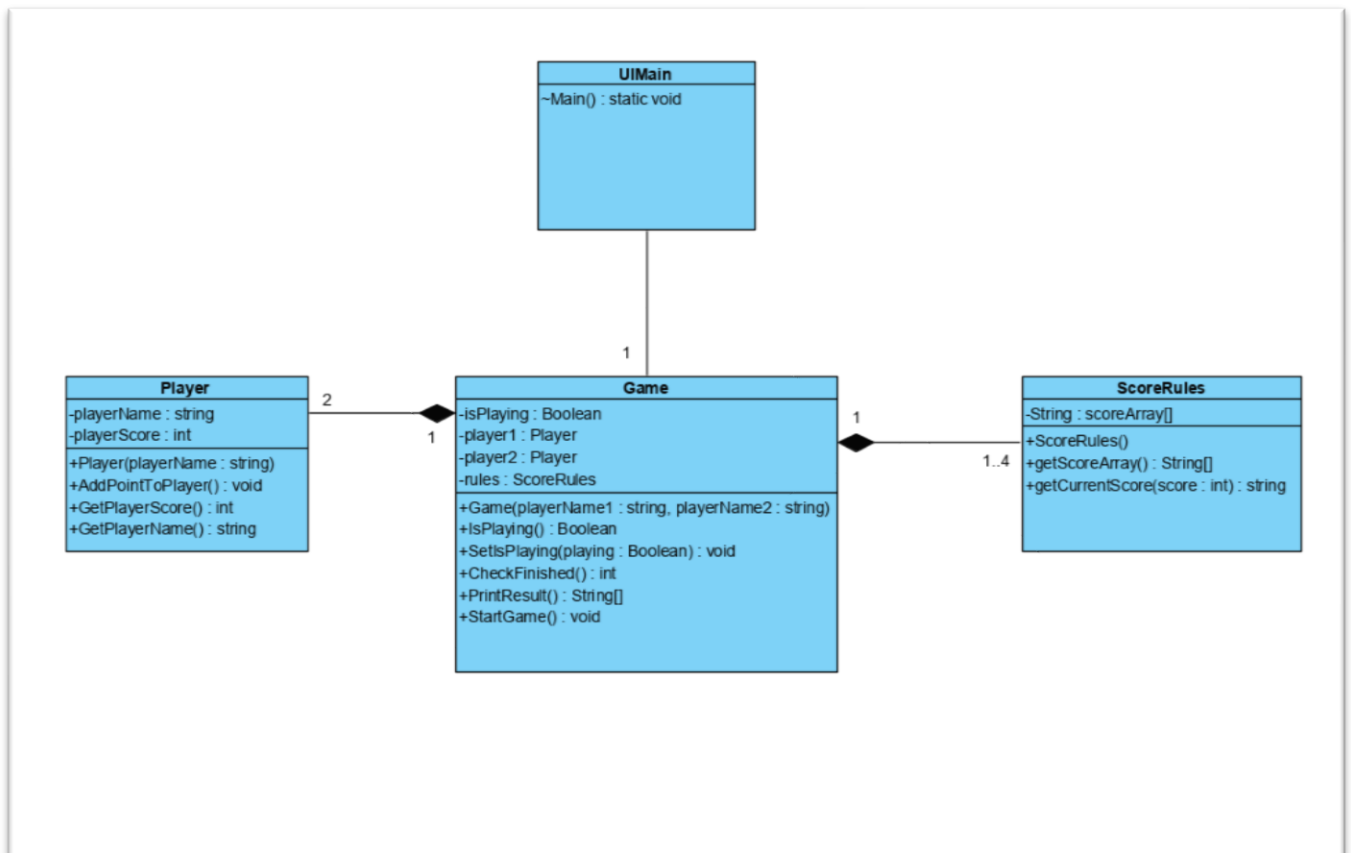


Figure 1: UML class diagram over the application

The design has been based on the SOC (Separation of concerns) where the **Game** class is responsible for the simulation of the game logic. **Player** is responsible for the score the player has received and name. The **ScoreRules** class has the array of the defined score rules based on tennis score rules.

If we considered implementing the whole tennis game, then it would be a good idea to separate the concern of rules to a separate class called **Rules**. The **Player** and **ScoreRules** classes have a high cohesion because the classes are focused on what they should do. The **Game** class also has a high cohesion because it is the general simulation of the game logic and therefore it has only the necessary methods in order to simulate.

I have taken into consideration moving some of the methods in the **Game** class, mainly the ones that define logical moves in tennis, to a class generally known as **Rules**, but I have decided not to do it, as the game has a better cohesion if it also has the general rules.