

## **Peer review on Daniel Sundbergs group (ds222me)**

### **By Fredriks Group**

**Try to compile/use the source code provided. Can you get it up and running? Is anything problematic?**

good

**Test the runnable version of the application in a realistic way. Note any problems/bugs.**

good

**Does the implementation and diagrams confirm (do they show the same thing)? Are there any missing relations? Relations in the wrong direction?**

The stand function is not implemented like the sequence diagram.

**Is the dependency between controller and view handled? How? Good? Bad?**

Nope it is still a hidden dependency between the controller and the view since the controller depends on the view to give the inputs to work.<sup>1</sup>

The controller should be the one getting the inputs.

**Is the Strategy Pattern used correctly for the rule variant Soft17?**

The soft17 rule is used correctly to our knowledge in this implementation.

**Is the Strategy Pattern used correctly for the variations of who wins the game?**

The strategy pattern for who wins is implemented correctly. The win conditions can be changed by making a new WinStrategy and not having to modify the dealer class.

**Is the duplicate code removed from everywhere and put in a place that does not add any dependencies (What class already knows about cards and the deck)? Are interfaces updated to reflect the change?**

Duplicated code has been removed in the rules but there are some duplicated code in the dealer also which should be removed, and NewGameStrategy should not be abstract when it does not need to.

---

<sup>1</sup> <http://tutorials.jenkov.com/ood/understanding-dependencies.html#visiblehidden>

It should be an interface, getting rid of the duplicated code is not its purpose. This can be done in dealer class without having any more dependencies.

### **Is the Observer Pattern correctly implemented?**

The observer pattern is implemented correctly.

But it is done using 2 classes, player class and game class when it should be done using one class when possible for the reason of Single responsibility principle<sup>2</sup>

we recommend to do it in game class, so it sends the message when hit or stand methods are called.

### **Is the class diagram updated to reflect the changes?**

The class diagram is updated and fits the new implementation, we can't find anything wrong else than it is a bit messy.

**Do you think the design/implementation has passed the grade 2 criteria?**

**The implementation of patterns has definitely passed the passing grade.**

Minor changes and then it should pass.

---

<sup>2</sup> [https://en.wikipedia.org/wiki/Single\\_responsibility\\_principle](https://en.wikipedia.org/wiki/Single_responsibility_principle)