A peer review by Johan Gudmundsson for WS3 made by the mr222yf

How is dependencies between the view and controller handled?

The dependencies between the view and controller are as they should. In other words your are letting the controller handle different scenarios by having dependencies to the view which is exactly how it should be done according to what Tobbe has said in the course lectures[1].

Is the Strategy Pattern used correctly for the Soft17 rule?

The Strategy Pattern is correctly used. And uses a similar implementation as the original code. You are clearly a lot better at Java then I am so It is hard for me to see any improvements since I don't fully understand your implementation. But a thought is, what would happen in a scenario where the dealer has 2 aces on hand? Would your implementation of the rule still work or would it break?

Is the Strategy Pattern for win variations correctly implemented?

The Strategy Pattern is correctly implemented, in fact it is the exact same way as I have implemented it. And as far as my knowledge goes this is the most optimal way to implement it, and I can't see any room for improvements at all.

Is duplicated code removed from everywhere?

You have successfully removed all duplicate code that I could find. In fact you also here has almost the exact implementation as I have. But with one difference, you have removed the original dependencies in the GameStrategy classes and instead added new ones. The assignment did not say anything about if that would be right or wrong. But the instructions for the peer review clearly states it. This is also not something that we have talked about at the course lectures, and not something that the book takes up either. In fact we have not talked much about code refactoring at all besides that it means to “rewrite code and keep the same functionality”. So I can’t honestly say if your refactoring is correct or not, I was also not able to find any good resources/references for this online. But I suspect that this is wrong, and I would suggest that you take this up with Tobbe before sending in the final submission.

Is the observer pattern correctly implemented?

At first I was very impressed with how you had implemented Observer pattern. But then I noticed that you have not used any observer interface so I actually think your implementation is incorrect. Because both when Tobbe show a implementation of the observer pattern in the course lectures[1] he makes use of an observer interface. and also gofpatterns[2] shows a class diagram where you would make use of an interface. I would like to add that there might be other ways to implement an observer pattern that I am not aware of, I'm just referring to what I have learned during this course, so in the end your way of implement the observer pattern might be correct as well.

Do I think you have passed the grade 2 criteria?

It does. I think all of your work follows a very high standard. And I can't see any way why it should not. The only thing I think you should do is to reach out to Tobbe and ask if it is okay or not to change dependencies during refactoring. Other then that I would say you have done a great job!

References

1. Course Lectures by Tobbe
2. Behavioral design patterns (no date) Available at: https://www.gofpatterns.com/behavioral-design-patterns/behavioral-patterns/observer-pattern.php (Accessed: 31 October 2016).