Workshop 3 Peer Review – Ola Franzen

Try to compile/use the source code provided. Can you get it up and running? Is anything problematic?
Program works perfectly. No issues found

Test the runnable version of the application in a realistic way. Note any problems/bugs.
Very slight and erratic bug; sometimes, due to my impatience, I would press "h" before the final card had been dealt. This then resulted in the game ending and no card was given. This was most commonly found if one card was given to the player and none to the dealer, but it was rarely found.

Does the implementation and diagrams conform (do they show the same thing)? Are there any missing relations? Relations in the wrong direction?
No, the diagrams and implementation are both in agreement.

Is the dependency between controller and view handled? How? Good? Bad?
The controller class "PlayGame" is dependent on the view as it has it as a function. The view is totally independent with no external dependencies. This is a well handled dependency system.

Is the Strategy Pattern used correctly for the rule variant Soft17?
Yes. The Strategy Pattern requires a single interface for the strategy, which can then be utilised as soft17 or the standard rules as required. As there is still the basic hit strategy and a new class of this strategy "Soft17HitStrategy", it is correct.[1]

Is the Strategy Pattern used correctly for the variations of who wins the game?
Yes. As per above, there is a single interface with multiple variants, one for dealer winning on a draw and one for the player winning on a draw. This obeys the Strategy Pattern as they are now independently interchangeable from the client function.[1]

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?
Yes. As the dealer already knew about the cards and the deck, no further dependencies were added. All of the interfaces appear to be correct.

Is the Observer Pattern correctly implemented?
Yes, a subscriber alerts the view that an event (the dealing of a card) has happened and resulted in a change to the view.[2]

Is the class diagram updated to reflect the changes?
Yes, the class diagram reflects the code well.

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

Yes, all the above criteria are very capably met.

1. Gamma, Helm, Johnson & Vlissides (1994). Design Patterns (the Gang of Four book). Addison-Wesley. ISBN 0-201-63361-2
2. http://www.oodesign.com/observer-pattern.html