

Workshop 3

Peer review for Kristoffer Lind, Hannes Ljusås, Anton K Andersson

By Juhani Aavanen and Eddy Proca

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

The code compiles and the app runs without any problems.

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

There are no major bugs. The only strange thing we could find was that it's possible to enter hit or stay on the start menu before starting a new game.

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

The implementation and diagrams conform. The only missing relation is to RulesFactory. There are no relations in the wrong direction.

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

The game and view are sent and declared each time the Play method is used. This could be done in the constructor instead. The view contains the Action enum which isn't specific to the view and therefore should be placed in the controller. Otherwise, everything looks good.

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

The rule variant Soft17 is implemented according to the Strategy Pattern, but some code has been duplicated in the DoHit function instead of reusing the code found in Player to calculate score.

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

The variations of who wins the game is implemented according to the Strategy Pattern. However, there is a line of duplicated code regarding the max score which isn't really relevant to the tie rule.

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?

The duplicated code was removed and replaced by a single method in the Dealer class, which causes no dependencies. However, the new game interface and strategies still have a deck argument which is no longer used.

Is the Observer Pattern correctly implemented?

The subscribers are connected to the Player instead of the Dealer (who is the publisher). This results in using the observer pattern on every object of a Player instead of just the Dealer. The

controller is notified that a Card has been dealt, and sends this as an argument. However, that Card is never used; instead all cards are rendered every time. The Observer Pattern is implemented, but could be a bit more effective with a few tweaks.

Is the class diagram updated to reflect the changes?

The class diagram has been updated and reflects the changes made to the code.

Do you think the design/implementation has passed the grade 3 criteria?

The design and the implementation is of high quality and it easily passes the grade 3 criteria.