

# **Peer review**

On design/implementation of Pär Eriksson and Oskar Emilsson

## Questions

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

We did not have any problems with compiling the source code.

A small recommendation about the code is that, we were asked to implement the operation `Game::Stand`. The for loop inside the `Stand()` is the `showHand()` method from the player class.

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

The class diagram that was submitted does not seem to have any faults.

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

The hidden dependency in the controller and view has to do with the low level inputs that the controller receives from the user. These inputs must be handled by the view and not the controller. In the book in section 17.13 it is mentioned, that classes like view “should not fulfill tasks associated with ; they typically receive these events and delegate them to a controller” [Larman 1, p.448]

Although this team used enumeration inside the view interface, this does not change the fact that the controller continues to be aware of these low level inputs and stores these characters inside the controller and then compares them to the enumerator fixed values.

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

The general logic is correct. The interface is correctly implemented. But when the dealer has an ace and two other cards and you have a total sum of 17, you should be able to hit and this is not happening since we have the limitation with the number of cards.

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

Yes, everything seems to work well.

6) 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?

We can see that the duplicated code is removed from the dealer class but the `AmericanNewGameStrategy` and `InternationalNewGameStrategy` still have the code.

7) Is the Observer Pattern correctly implemented?

We are not experts in patterns and we cannot say for sure if everything is done correctly. It is strange that someone can subscribe inside the game class which suggests that the game class is the publisher but the events are published by the player class.

8) Is the class diagram updated to reflect the changes?

Yes, the class diagram is updated.

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

Generally, it was a good effort from this team. We believe that if above faults are corrected this design can pass the grade 2.

**References section.**

- 1) Larman C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0131489062