

## Peer Review

For: 1DV607, Workshop 3, Grade 2

To: Patrik Nilsson(pn222en)

From: Rasmus(rl22dv), Henry(hp222fq), Seldin(smubc09)

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

- We got the code up and running without any problems.

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

- We tested the code. While not all the requirements for Grade 2 was implemented the program ran fine without any problems.

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

- We could not find any UML class diagram in the link you provided us with unfortunately.

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

- The hidden dependency is handled. Although it can be made much simpler and will reduce your code slightly (by using a value of some sort instead of creating methods for each scenario play, hit and stand).

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

- Soft17 is implemented while hasAce field is not mandatory in this solution. You can get away with a less code, but this is just picky and one could say this adds readability. Otherwise noting else to mention.

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

- Win strategy is implemented correctly.

**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 is still there, it seems that the group has completely forgotten about this. Or maybe they misunderstood, but the part

```
c = deck.GetCard()
```

```
c.show(..)
```

```
player.DealCard(c)
```

is duplicated both in AmericanNewGameStrategy and InternationalNewGameStrategy and should be logically removed. One could argue about the same code in Dealer but we don't.

**Is the Observer Pattern correctly implemented?**

- Observer pattern not implemented. The Interface is created and implemented in PlayGame but the code is not implemented, the implementation of the Observer pattern is not used completely correct. But you are in the right direction to solving the problem.

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

We did not find an UML class diagram as mentioned above, so no its not updated.

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

No we don't think you have passed the criteria for grade 2. Some requirements are implemented but others are not. You probably already know about this.

**To think about**

Nothing to remark on code wise just need to fulfill all the requirements.

**Not implemented or not implemented correctly**

: No UML class diagrams

: Duplicated code is not implemented correctly.

    c = deck.GetCard()

    c.show(..)

    player.DealCard(c)

: You just need to finish up the Observer pattern. Now it's 50% done.

**References**

1.Larman, C., Applying UML and Patterns 3rd Ed, 2005, ISBN: 0-13-148906-2