

Changes in source code

- Added Abstract Factory
- Game class takes argument with the game settings that are to be used (SneakyDealer)
- Added boolean to Showed Hand in player to know if card was changed from hidden.
- Added Visitor pattern, all rules can be accepted and if visit a “print line” will be triggered.
- Added code in the views to make the rules print out s

Visitor

I think the visitor has many drawbacks, especially when it comes to low coupling. So many more dependencies. But at least the Model-View separation principle could still be fulfilled which I guess is the point.

Abstract Factory

The Abstract Factory was really sweet. However in this small example with 3 rules with 2 variations the amount of classes needed don't favor the factory. But if added a couple of more rules and variations the needed classes would favor the factory. But anyhow it was nice to have a class responsible for creating the rules which would make less error prone if other persons would take over the code.

Class Diagram

As mentioned before both the Abstract Factory and especially the Visitor has made very strong coupling in the design, this is shown very clearly in the class diagram which now looks very messy even though some dependencies relations is shown by just an arrow to the package instead of one arrow to each class.