

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

Did not know how to set up the project, so could get it up and running.

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

Did not know how to set up the project, so could get it up and running.

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

Yes.

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

Yes, it's implemented with use of matching enumerations.

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

Yes.

"Larman states[1,p447] Define each algorithm/policy/strategy in a separate class, with a common interface[2]"

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

Yes.

7. 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.

8. Is the Observer Pattern correctly implemented?

Yes.

"Larman states[1,p465] "Define a "subscriber" or "listener" interface. Subscribers implement this interface. The publisher can dynamically register subscribers who are interested in an event and notify them when an event occurs." [2]"

9. Is the class diagram updated to reflect the changes?

Yes.

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

Yes i do.

### **References**

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