The first impression of your project seems very good. No instructions are given on how to run your source code, but you have a fully working application of your project.

Regarding your project, there are a few things worth mentioning that could be better:

--- 1 ---
Your source code could use some more formatting in order to make it easier to read it. Why use bracket sets if for example an if-statement or while-statement only has 1 function? Also, try to reduce your lines of code by placing your brackets better whenever possible.

--- 2 ---
Lack of comments.

--- 3 ---
Your class does implement an observer pattern. However, make it clear to the user through comments which parts and perhaps how the observer is used and implemented. Your observer patterns comes from through the CardDealtListener interface which is used in the PlayGame class. Regarding critic point 2, comments could explain a lot.

--- 4 ---
Within the “view” packages, try to use the bool method from the Card class IsHidden instead of comparing it to the enum value “Hidden”. That would make code shorter.

--- 5 ---
No instructions are given with your uploaded project. Provide a proper readme file if it is not clear on how to open, import, read or other related actions in order to review your work. For example which software to use and install to be able to see your work and so on. Most likely, a C++ development environment is required. But as part of the course, some users could opt in to use Java instead.

--- 6 ---
The class PlayGame checks for score values before call methods to print out the score. Try to implement this check within the method call itself, instead of within the class PlayGame. It makes the class longer than needed too, and it does not really seem the task for PlayGame to check that.

--- 7 ---
Some common code parts could be improved, such as having final attributes, using @Override (or similar in C++) and so on. Most development tools give warnings or suggestions on how to write code. This is due, these improvement suggestions are never used at all.

A must-have to fix, because of the requirements of the assignment:

--- 1 ---
A requirement was to document all changes made within the original source code. This was not

done at all. So it is hard to know, what you exactly did change.

All things considered good for your project would be:

--- 1 ---
Input works responsive.

--- 2 ---
All non-discussed requirements within this report are executed well enough.

--- 3 ---
Nice presentation, without any oblivious mistakes.

--- 4 ---
Lot of new created files within the “rules” package which uses good in-line naming with other existing files, including with interfaces. You also did properly include an extra implementation for the RulesFactory class regarding max score.

--- 5 ---
The class PlayGame is used correctly, save for some checks that should be preformed elsewhere. The method Play still remains an bool method, so it saves adjustments for the Program class (which in fact is almost untouched, which is even better).

--- 6 ---
All requirements, except commenting are done well enough.

In short:
Your program would require commenting documentation in order to show what you exactly did change. Documentation on how to use your program is recommended. Also, your program will benefit a lot from formatting. The original source code that was given was of course not properly formatted, but still.

The bottom-line:
Enough for grade 3? Yes, I think it would suffice. The only required thing left to do is to document all changes. Other critics are not part of the grade, but could benefit you in order to make it easier obtain grade 3.