**FOOP HW3- Old-Maid Re-implementation and 2 Variants**

B01902039 施楷文

1. Introduction

In this homework, I modified the program in hw1, making it more object oriented. And there are two variants I made. **The first one** is the basic variant that is on the wiki page, in the game, there are no jokers and there is randomly one card been removed from desk. **The second one** is, there is randomly one card value (2, 3, 4… J, Q, K, A, Joker) that these cards can only be dropped when they matched the value and color (Club = Spade, Heart = Diamond). The other rules remain the same.

1. My design of class structures

Main class:

POOCasino

Hand

Player

Player

Player

Player

Computer

Card

Card

Card

Card

Random

Index

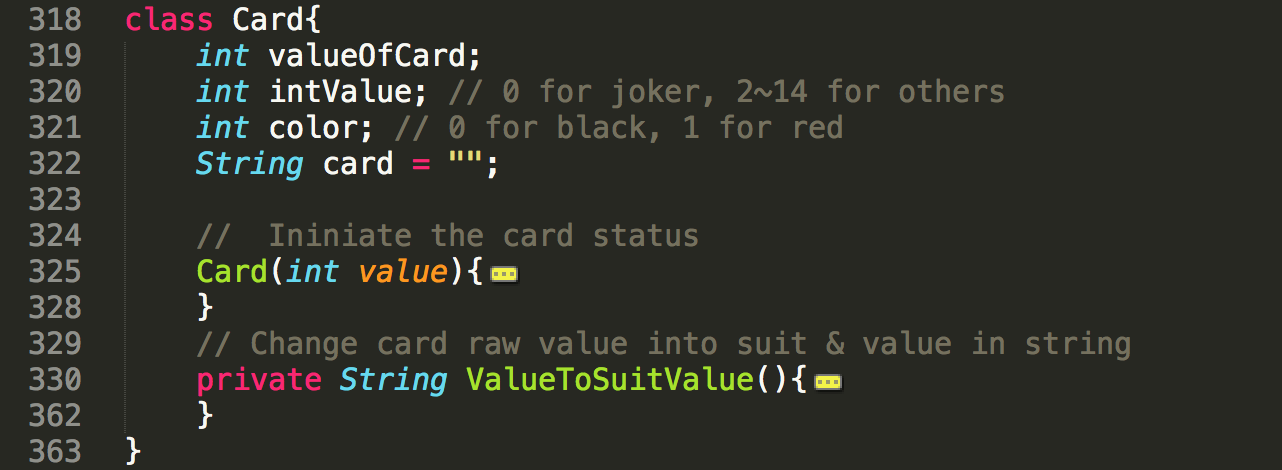
Main class- PlayGame: Call the computer class including the original version and two variants to start and host the game.

Computer class (including the two variants): The computer class have to maintain the game status & the players’ status, start/host/stop the game, and the most important job is to communicate with the player.



Player class: The class contains the player status, what cards does he have, and also some methods manipulating cards in hand.

Card class: The card class contains the status of one card, including card value in several forms (integer: -2~51, 0 & 2~14, string such as “C10” “SA”)



Shuffler class: I use the class provide by TAs in hw2.

1. How to play with the two variants?

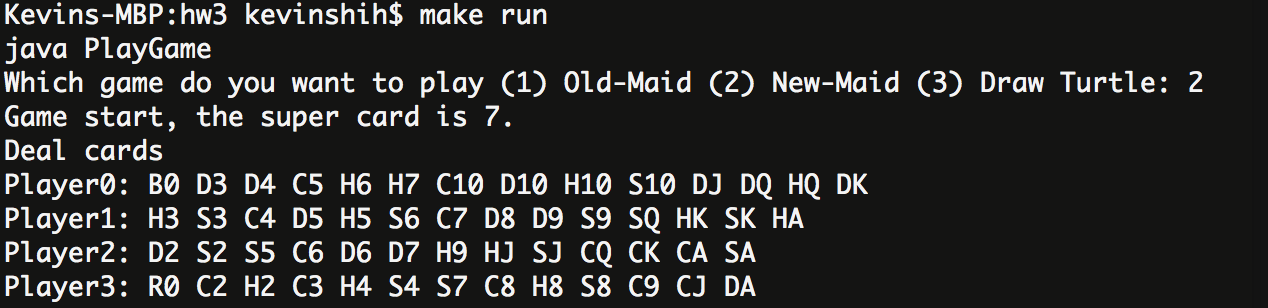
Since there are no big differences, the way to play with my program is quite similar.

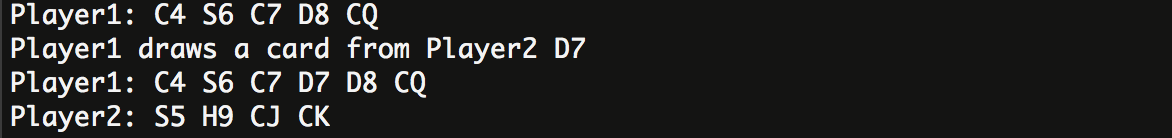
1. Open command line.
2. **$make all** in the game directory.
3. **$make run** in the game directory.
4. Type in which game mode you want to play, the rules of the two variant games are quite trivial.
5. How I tested my program?

Since my hw1 program was totally tested, I only modified the structure of the program to make it more object oriented. In this homework, I use the test program (which is provided by my classmate Yeng-Chieh Sung) which is the one I used in hw1 to make sure there are no mistakes in game. And then, to make sure the variants work, I manually manipulated the game to make sure if it follows the rules. I simply add some codes to receive input in order to cut every round separately that I can read the output more clearly.

1. The sample output of each variant

Variant 1:





Variant 2:

