The Big Two

Project Background and Description

My project goal is to Java to create the popular Asian card game the Big Two.

The main basis of this game is a race between players to get rid of their hands as soon as possible. It supports 2 -4 players with a full poker deck.

The way of getting rid of the cards is by putting a stronger card than the previous player in specific sequence for each round.

To compare the cards, we first compare the number, then the suit.

Number from strongest to weakest is the following:

2>1>K>Q>J>10>9>8>7>6>5>4>3

Spade from strongest to smallest is the following:

Spade>Heart>Diamonds>Club

In a card match, 4 will always be stronger than 3 disregarding their suits. However, Spade 3 will be stronger than any other card of 3 in game. So, as a result, Spade 2 will be the strongest single card in the game, and Club 3 will be weakest.

There are 6 types of sequences, and which card we compare of the same sequence:

Single (just one card, strongest card)

Pairs (two cards of the same number, the strongest card in that pair)

Straight (five cards of consecutive numbers, the strongest card)

Full House (3 of the same number plus pair, the strongest card in 3 of the kind)

Flush (5 cards with the same suite, the strongest card)

Four of a kind (4 of the same number plus a 1card, the strongest card in 4 of the kind)

The player who holds Club 3 get to get rid of their card first and use it to decide the sequence of the first round to start the game. Player must only get rid of cards in the same sequence in every round. Player may pass if they have nothing to get rid of, and the one who put out the strongest sequence in that round get to decide the sequence of next round. The player who get to get rid of all their card first wins the same.

2.

Project Scope

I will be the only member in this team and will be solely responsible for the completion of this project. The completion of this project is determined by if players are able to finish a full game of the big two and provide a clear winner meeting high-level requirements.

3.

High-Level Requirements

The new system must include the following:

• Ability for each player to register with the game.

• Ability for the game to communicate that the game has ended.

• Ability for the game to communicate the winner of the game if the game ended.

• Ability for the game to determine the starter of each round.

• Ability for the game to know if the player got rid of a valid card sequence determined each round.

• Ability for players to know the card in their hands at all time.

• Ability for players to the sequence of the current round at all time.

• Ability for players to know the current number of cards of other player’s hands at all time.

4.

Implementation Plan

The code will be stored in the following Git repository URL:

<https://github.com/palmswill/CardGame>

The code will be reviewed twice a week, precisely every Wednesday and Sunday.

Text files will be stored under document folder, code in the source folder and a UML diagrams in VPP folder.

I will be following OOP FOR Java and the Java naming convention. Coding IDE and UML planner and code editor such as Visual Studio code, NetBeans and visual paradigm will be used for this project.

Design Considerations

•

Encapsulation

Current player should only be able to see the number of other players but not the exact hands. Array list of hand for each player will be set to private and the getter will only be called when the player is active.

•

Delegation

The game class will call on the round class to start the same each round. It will also get if the game ended and the player who starts the next round.

There will be a card distribution class called by game class that handle card distribution.

•

Flexibility/Maintainability

Card distribution class will be separated from the Enum class definition content of card deck. So, if rules change to multiple deck (the big two can also play with two decks) in the future, new deck can be inserted into the card distribution class for card distribution to player.

toString method for all class will be rewritten to better present the object.