Class Extended Brainstorm

Player:

Private Card[] hand

Private int points

Card:

Private String name

Private String suit

Private String value

Private int pointValue

Deck:

Private Card[] deck

Public Deck()

(initialize all the cards)

Private void shuffle()

Game Restrictions (Might be part of game screen controller):

Private String Difficulty (for single player)

Private String NumberofPlayers

Private int pointLimit

Method for choosing right card depending on suit played

Jack of Diamonds rules

Q-Spade breaks or not

When Hearts are broken which allows hearts to be played

Music:

TBA

Things to think about:

How to regulate rounds

How to distribute the cards from the deck to the player’s hands

How to dynamically change the size of the player’s hand of cards as well as the gui that represents each player’s hands

Game handling:

* Should I have a central pile class that takes in 4 cards each round and determine rules from this class?
* Should I have the game controller do the logic?

13 rounds total…

Should each player also hold a discard pile… (I think this is a good idea). Then from this discard pile, we can extract cards that have points, and these will determine each player’s points.

Should I have a score class?

Or is score keeping for each player enough? (I think so)

-Make sure the indicies in deck and other classes will not exceed 52.

Game Regulation

Organize cards based on suit and value

Design cards

Design UI

**Game Regulator:**

Created inner class to hold association between Card and who played it.

Calculate the highest value, and assign the right pointer to the player

To determine the points we must consider the highest value of the card of the same suit that has been played by the first player….

1. Calculate who needs to play next
2. Determine points
3. *1.5 GameRegulator changes: Accept card method where it will add cards to a container  
   \* within Game Regulator class. Created inner class CardIndexContainer that keeps track of the  
   \* Card and who plays it. Determine the amount of points allotted to the player that plays  
   \* the highest card. Made method that determines who plays next, depending on the selection of the cards.  
   \* GameController: Added functionality that only allows the player to play their cards  
   \* during their turn only.*