Matthew Pribadi
W200 – Benoit – Mon 6:30pm
Spring 2021

Phỏm Card Game Reflection

Overall, I believe I completed a reasonable implementation of the Phom card game as used in a terminal, text-based format. To go about testing this game, I would recommend first reading the rules in this link. Interacting with the game is very simple. You can play the game by yourself as four separate hands, or you can take turns passing it along to another person in the room. I have it set so that after each turn, the previous hand's moves are cleared so that the current player is not tempted to look at the previous players' hand. After the introduction, the game immediately begins with the first player's turn (having automatically drawn a card to begin the game to start with 10 cards). From there you have the option to discard one of the 10 cards. The next player will then have a selection of actions they can take that are well commented. You can test the game with all sorts of inputs, but it should keep you in the game until you enter a valid input. After four rounds the game will end, and scores will be calculated.

With this game implementation, I was able to complete all the basic gameplay mechanics of the game, specifically around "hitting" and "melding". I implemented the idea that if a previous player were to "meld" a card, they no longer have the ability to discard that card (meaning they are bound to the "melded" card). A mechanic of the game I was not able to implement revolved around melding a card and not completing a set. In the real game, you cannot meld a card that you cannot form a matching set with immediately. Another implementation that I was not able to complete was that on the final round, a player has the option to push any cards in their hands to any previous player who has revealed all their matching sets (indicating the end of their four rounds).

In beginning to code this project I thought that, overall, it was a simple idea to execute as it was an extension of the playing card homework, that we had done in week 7. However, as I began to implement some of the basic mechanics of the game, I ran into multiple issues. The prominent issue I had was in keeping track of terminology and variables. More than once, I had used the same name for a class variable and a method, thus resulting in errors that took me hours to troubleshoot. Another issue is keeping track of mutable and immutable variables. I had to understand where my class objects were being manipulated or getting updated and had variables changed several times when I did not want it to be changed. Further improvements to this code would be to utilize a better clear screen option, using Unicode or images for the cards, and a better GUI. I would also move all the scripting out of the final main statement and utilize more functional programming within the object Class (a better laid out system for user entry and menu commands).

References: https://www.vinagames.com/phom.php

Card Point Values:

Card	Points								
A	1	**4**	4	**7**	7	**10**	10	**K**	13
2	2	**5**	5	**8**	8	**J**	11		
3	3	**6**	6	**9**	9	**Q**	12		

Final Implementation Structure

Orange – added|| Red - Modified

class PlayingCard:

'''Creates a playing card instance with a specific suit and rank'''

def __init__(self, value, suit)

def getValue(self)

def __str__(self)

Properly displays a card for use in playing the game.

def __repr__(self)

Properly displays a card for use in playing the game.

def __lt__(self, other)

def __gt__(self, other)

def __eq__(self, other)

def __le__(self, other)

def __ge__(self, other)

def __sub__(self, other)

class Hand:

'''Instance of all the hand actions one can perform when playing a card game.'''

def __init__(self, cardList)

def dispMatches(self):

Displays number of current matching sets in the hand.

def orderCards(self):

Orders cards to increase readability.

def displayHand(self):

Displays the player's current hand.

def getDiscard(self):

Displays the player's discard pile.

def popDiscard(self):

Displays the player's discard pile.

def scoreHand(self):

Adds up the number of points in the player's hand.

References: https://www.vinagames.com/phom.php

```
        def __str__(self)
                Properly displays a hand.
        def __repr__(self)
                Properly displays a hand.
class Deck:
''' Creates a deck of 52 playing card instances with no repeats of one'''
        def __init__(self)
        def shuffle(self)
                Shuffles the deck at random to prepare for playing a game.
        def deal_card(self, cardCount = 1)
        def draw(self, num=1)
                Draws a set number of cards based on the num argument entered.
        def __str__(self)
class CardGame():
''' Creates a game instance that utilizes a deck of cards '''
        def __init__(self, playerCount):
        def deal (self, num):
                Deals a 'specific number of cards to the number of players.
        def dispCards (self):
                Displays all the cards for all players in the game (for troubleshooting)
        def incRound(self):
                Increments the number of rounds played.
        def getRound(self):
                Returns the current round.
        def deckLeft (self):
                Returns the number of cards the deck has left.
        def gameEnd(self):
                Placeholder method for a game end condition.
        def __str__(self)
class Phom(CardGame):
''' Creates a Phom game instance that extends a type of CardGame with extra functions'''
        def __init__(self, playerCount):
        def dispChoice(self):
                Displays the previous opponent's top discard card.
        def dispPlayed(self):
                Displays all the discarded and melded cards.
        def dispCurrentHand(self):
                Displays all the discarded and melded cards.
        def hit(self):
                Calls the parent class's Draw function.
```

References: https://www.vinagames.com/phom.php

```
        def meld(self):
                Takes the opponent's top discard card. Calls roundCounter. Reorganize the
                discard pile size.
        def discardOne(self):
                Calls the parent class's Draw function.
        def gameEnd(self, index):
                Checks for the game end condition. Then calls Hand.dispMatches and
                Hand.score for that player.
        def finalScore(self, Hand):
                Displays the final score and announces the winner.
        def gameEnd(Phom):
                Returns True if the game end condition is reached (four rounds or a complete
                matching hand)


if __name__ == "__main__":
        # Initialization and displays the current game progress
```

References: https://www.vinagames.com/phom.php