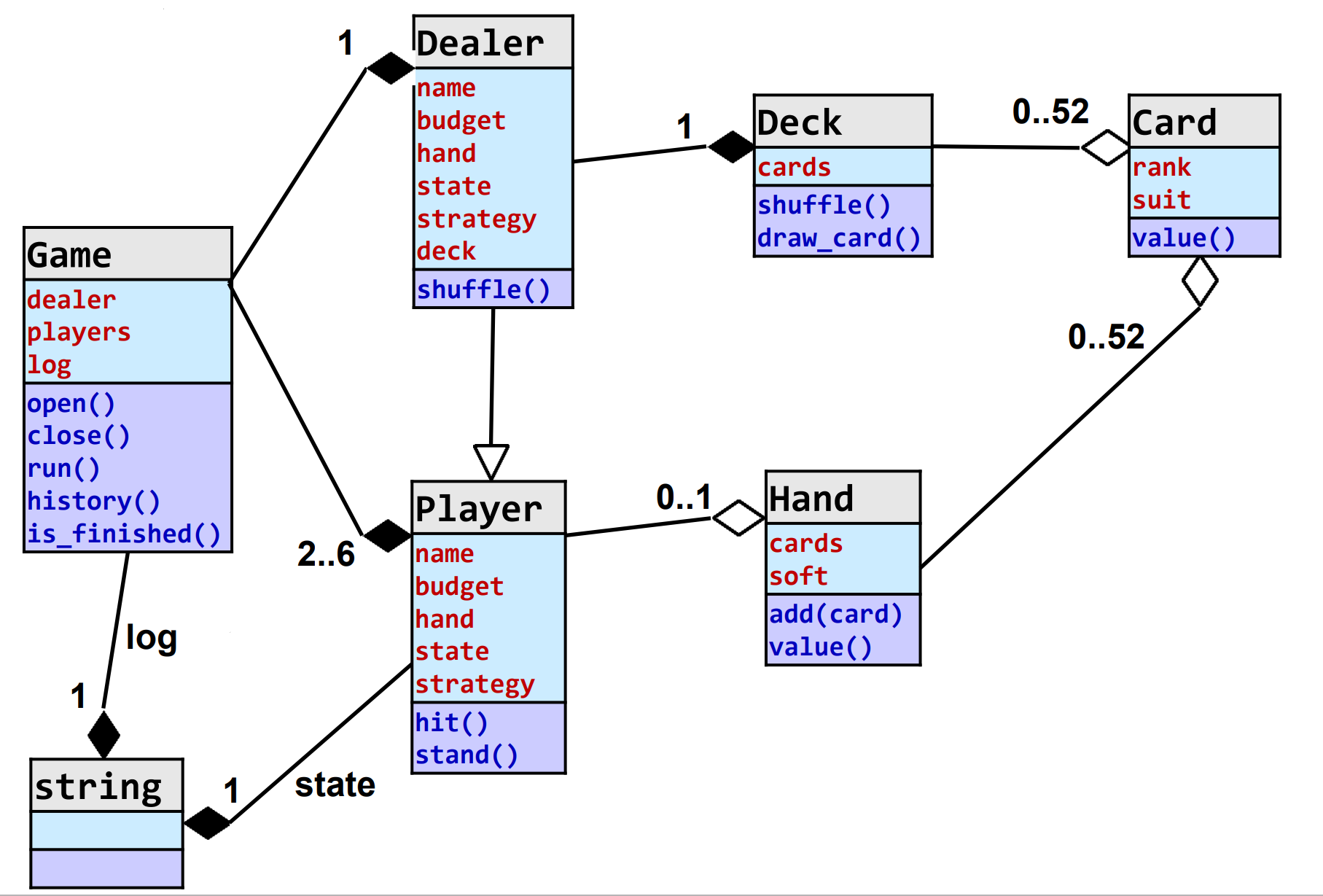
**FINAL PROJECT DESCRIPTION (Team of 2 Students)**

1. You will create an account on GitHub for your project repo and add me id: phenyle
   1. I will track all the code commits and changes so I can gauge your participation during the next 2 weeks
2. You will also upload the UML class diagrams to the project repo
3. There will be no submission via canvas, your repo is the submission site and the deadline is Dec 4th
4. On Dec 4th the team will present the class diagram and code and demo the project during class

**Description:** In this project you will implement two popular card games (Blackjack and Texas Hodlem) as your final project. Texas Holdem will be an extension of the code written for Blackjack and is therefore for extra credit. The project team should submit the UML class diagram for Blackjack, implement the code based on the UML diagram and be ready for demo and presentation on Dec 4th. For extra credit the team can submit the UML diagram for Texas Holdem and the code implementation.

**Blackjack rules:** The game is to be played between one player and the dealer (computer player). Below is a simple UML example for Blackjack to get you thinking in the right direction. You will need to create your own detailed UML based on the rules.



* The object of blackjack is to get a hand of cards as close to a sum of 21 as possible without going over. If you go over 21, you bust and lose the hand automatically.
* Before cards are dealt, you must place your bet. This money is immediately subtracted from you bank roll. Bets are always $100 in our implementation. The player starts with $1000.
* Two cards are dealt to both the player and the dealer, one face up and one face down. You can look at both your cards, but can only see the dealer's face up card. The dealer can see both his cards. Number cards are worth their face value. Jack, Queen, and King are worth 10. Ace is worth either 1 or 11, whichever gives you the better hand. An ace may change value during the game (i.e. if you have an ace and an eight, a sum of nineteen, and then draw a five, the ace will change to be worth 1 so you don't bust).
* If you are dealt a sum of 21 on your first two cards, you have a blackjack and you automatically win 2.5 times your initial bet. If both you and the dealer have a blackjack, the game is a push (tie).
* You may choose to hit (draw a card) or stand (do not draw any more cards). If you hit, draw a card from the deck and add it to your hand. You may continue to draw cards until you decide to stand or you bust. Once you decide to stand, the dealer draws cards. If you bust, you lose automatically and the dealer does not draw any cards.
* The dealer must continue to draw cards until he has a sum of at least 17. Once he has a sum of at least 17, he stands. If he goes over 21, he busts and you win.
* If neither player busts, whoever has the higher total wins. If both totals are the same, the game is a push.
* In the case of a push, you get your initial bet back.
* If you win, you win double your initial bet.
* Continue to play until you are out of money or decide to quit.
* If your initial two cards have a sum of 10 or 11, you may decide to double down instead of choosing the normal hit/stand options. If you double down, you must wager an additional 0.5 times your initial bet ($50) and you must immediately draw one more card. You may not draw any more than one card. The dealer then plays as normal. If you win, you win 3 times your initial bet (twice your initial bet plus twice the additional 0.5).
* Don't worry about splitting.

**Project Requirements:**

* The game begins with **Five** standard 52-card deck shuffled together to form the game deck. Cards are drawn from the top without replacement. You should not shuffle cards in the middle of a game, unless you take account the cards in the players hands. To avoid this problem, determine the maximum number of cards that can be used in a game and reshuffle between games if there are less cards in the deck than might be used.
* Implement the deck as a dynamic array inside the Deck class. Create a constructor for the Deck class that accepts the five 52-card decks to load. Cards are drawn from the high end of the array.
* The deck must be sufficiently shuffled at the beginning of the game.
* To aid in testing, you must give the option of printing the cards in the current playing deck in order. Print 20 cards per line, separated by spaces. Instead of printing "10" for a ten, print "0" so everything gets aligned nicely. Use "J", "Q", "K", and "A" for jack, queen, king, and ace, respectively.
* You can choose your own algorithm to shuffle the deck, or just swap two random elements 10000 times. You can use the function rand() to generate pseudo-random numbers.
* Input must be 'H' (hit), 'S' (stand), 'D' (double down; see example output), '?' (show deck), or 'Q' (quit).
* The player starts with $1000. The initial bet placed before each hand is always $100.
* Do not clear the screen after each hand. Just let it scroll.
* At a minimum, you should have two files for the Deck class, two files for the Blackjack class, and FinalProject.cpp. A generalized Deck will also have two files for a Card class. Note that none of the logic for the Blackjack game itself should reside in FinalProject.cpp.

**Sample Run**

Bank Roll: $1000

You bet $100. You have $900 left.

Dealer shows a Q.

Your hand: 6, 2

(S)tand, (H)it: ?

4 3 6 A J 8 3 K 3 5 J 5 9 J 0 K 8 0 4 K

9 4 5 9 5 0 8 J 5 2 7 Q Q 9 A 3 5 3 3 6

6 9 9 4 0 A 5 0 A Q 2 6 3 9 J 6 9 8 K 0

Q 4 6 6 4 Q 8 2 3 7 3 A K J J Q Q 2 K 0

8 J 2 J 7 9 2 5 2 K A 8 A A 0 9 2 3 7 5

4 7 3 K 2 5 J K 5 3 8 5 9 5 5 6 4 K K 6

9 7 0 9 Q 8 2 6 9 7 9 4 3 7 6 8 0 9 2 3

8 8 J K 7 5 5 8 3 8 2 0 2 4 6 6 4 J Q 3

A 8 Q 7 K 6 9 8 K Q A Q 3 3 8 4 A A J J

K A A 5 8 9 K J 4 8 2 4 2 J 5 7 5 6 7 6

3 2 Q 8 6 Q 0 4 K A 4 2 A 7 0 Q A 3 7 0

9 A K Q 0 2 0 J 7 4 2 J 4 6 Q 7 Q K Q A

K A 0 4 J 5 7 J 9 7 0 4 6 0 0 7

Dealer shows a Q.

Your hand: 6, 2

(S)tand, (H)it: H

You draw a 7.

Dealer shows a Q.

Your hand: 6, 2, 7

(S)tand, (H)it: H

You draw a 10.

You bust.

Bank Roll: $900

You bet $100. You have $800 left.

Dealer shows a 10.

Your hand: 10, 4

(S)tand, (H)it: H

You draw a 7.

Dealer shows a 10.

Your hand: 10, 4, 7

(S)tand, (H)it: S

Dealer draws a 9.

Dealer's hand: 6, 10, 9

Dealer busts.

You win $200.

etc.

**Poker**: Texas Holdem is a variation of Poker and Poker is played with a standard deck of 52 cards consisting of 4 suits and 13 ranks. Ranks in order from high to low are Ace, King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2. The suits are Diamonds, Hearts, Spades, and Clubs and are unordered.

Poker games generally consist of the following activities:

* Dealing cards to players
* Betting or folding
* Determining a winner based on the contents of the players' hands

What makes one poker game different from another is the details of these rules, such as how many cards are dealt, when betting occurs, etc., but there is some common terminology across all games. First, when it is a player's turn to bet, there are generally four options available (but some games may restrict when each option is available):

* Fold: A player who folds drops out of the game. He/she forfeits any bets made in earlier rounds, is not able to participate in any future betting, and is unable to win the pot.
* Bet or Raise a Bet: The player increases the amount that is being bet in the current round. To stay in the game, all other players, in their turn, need to bet at least this amount.
* Call: The player matches the current bet amount exactly.
* Check If no player has yet bet in this round, a player can essentially bet $0.

When the game ends, the player with the highest hand wins all the money that has been bet. Hands are ranked as follows, from highest to lowest:

1. Royal flush: Ace, King, Queen, Jack and Ten all of one suit
2. Straight flush: any five cards in sequence all of one suit. If there are multiple straight flushes, the one with the highest card wins.
3. Four-of-a-kind: four cards of the same rank, like 4 Jacks. If there are multiple Four-of-a kinds, the highest ranking one wins.
4. Full house: three cards of one rank and two of another. The hand with the highest three of- a-kind wins.
5. Flush: five cards from the same suit. If there are multiple flushes, the hand with the highest ranking card wins. If these are the same, the second highest is compared, and so on, to break the tie.
6. Straight: five cards in a sequence of mixed suits. In case of multiple Straights, the hand with the highest card wins.
7. Three-of-a-kind: three cards of the same rank. If there are multiple Three-of-a-kinds, the highest ranking one wins.
8. Two pairs: two cards of one rank and two cards of another rank. Ties are broken by comparing the rank of the higher pair, then the rank of the lower pair, and then the rank of the fifth card.
9. One pair: two cards of one rank. Ties are broken by comparing the rank of the pair, then the rank of the highest card not in the pair, then the second highest, etc.
10. High card: bad hand! If multiple hands have the same high card, compare the next highest, etc.

In any case where a tie cannot be broken, the pot is split evenly among those who tied.

**Texas Holdem Rules**:

Here are the sequences of actions for a single hand:

1. Dealer shuffles the deck. With each hand, the next player to the left becomes the dealer.
2. The player to the left of the dealer is required to bet half the minimum bet (before receiving any cards), called "posting a blind".
3. The next player to the left is required to post a blind equal to the full minimum bet.
4. Each player is dealt two cards, face down, called the "hole cards".
5. First betting round starts with the player third to the left of the dealer. Players can call, raise, or fold on their turn, but they may not check.
6. The dealer deals three cards face up in the middle of the table. These are called the community cards.
7. The second betting round happens, starting with the player to the left of the dealer. Players may check, call, raise or fold on this and all future betting rounds.
8. The dealer adds a face up card to the community cards.
9. There is a third round of betting beginning with the player to the left of the dealer.
10. The dealer adds a fifth face up card to the community cards.
11. There is a final betting round beginning with the player to the left of the dealer.
12. All the players who have not folded turn their cards face up. A player's hand consists of the five cards that create the best hand using the five cards in the community cards and the two cards that were dealt to the player. The player with the best hand wins the pot.

**Grading:** UML for blackjack (5 points), Implementation (10 points), demo and class presentation (5 points) Extra credit UML and implementation for Texas Holdem (10 points). You are currently at 90 points in your assignment total, with this project you will be at 110 points, I will remove the worst scoring assignment from (1-7) to a total of 100 points that your assignment score will be based on. If you complete the extra credit then you will get extra 10 points over your total grade for this class to boost your grade.