Team Members: Johnny Hsu, Justin LeBreck, Josh Ziegler, Michelle Ziegler

Responsibilities:

Johnny

- Dealer - Deals specific poker games
- PokerGameFactory - a Abstract Factory encapsulating and creating all factories,
- TexasHoldemDealer - A dealer responsible for hold'em-like game, like Texas Hold'em or Omaha
- DrawDealer - A dealer responsible for Draw game, like 5 Card Draw
- StudDealer - A dealer responsible for Stud game, like 7 Card Stud
- PlayerFactory - A concrete factory creating new human player object or AI player object
- TableFactory - A concrete factory creating new table object.
- DealerFactory - A concrete factory creating new Texas Holdem dealer object, Draw dealer object, or Stud dealer object.
- BettingStructureFactory: A concrete factory creating new No Limit object or Limit object

Justin

- Player - Either AI or human
- HumanPlayer - Tracks Name, Chips, Bet, and Hand
- AI Player - Tracks Name, Chips, Bet, and Hand
- Table - Contains the players actively participating in the current game.
- TablePot - Contains the amount of chips anteed or bet by players.

Michelle

- HandStrength - Compares the relative strength of Hand[]s
- PokerHand - Create a 5 card hand out of the cards dealt into ComunityCard and PlayerHand
- HandStrengthAnalyzer - Create the correct type of PokerHand
- PlayerHand- Creates a Card<> that represents cards dealt to each player
- CommunityCard - Creates a Card<> that represents cards shared by all players at the table

Josh:

- PokerGame - Superclass for the poker game
- 5CardDraw - Subclass of the PokerGame type hierarchy.
- TexasHoldem - Subclass of the PokerGame type hierarchy.
- 7CardStud - Subclass of the PokerGame type hierarchy.
- Omaha - Subclass of the PokerGame type hierarchy.
- GameType - Interface containing types of games: cash games, tournament games, etc. This interface is one of 2 connecting nodes for our bridge patterns.
- CashGame - Sublcass of the GameType type hierarchy.
- TournamentGame - Sublcass of the GameType hierarchy.
- BettingStructure - Interface containing types of betting structures: no limit, limit, etc. This is the second interface for the bridge patterns contained in our class diagram.
- NoLimit - Subclass of the BettingStructure type hierarchy.
- Limit - Subclass of the BettingStructure type hierarchy.
- Deck - Contains an ArrayList of Cards that can be sorted or shuffled.
- Card - Represents a physical card containing a face value and suit, the face value/suit can be displayed as both an int and string.

Design Pattern:

1. Bridge: We will use the bridge pattern when to structure poker tables as either tournament or cash tables and also to structure the betting pattern for each table as either no limit or limit poker.

2. Strategy: We use multiple strategies throughout the design. We have a poker game strategy that chooses between Texas Holdem, 5 Card Stud, 5 Card Draw, 7 Card Stud. We employ strategy to create players as either Human or AI. We use strategy for a list of dealers that corresponds to the list of poker games.

3. Template: We plan to use Java comparable library to compare different Card, PokerHand, and HandStrength objects.

4. Observer: The implementation of an observer pattern will inform players, after they sit at a table, that a card has been dealt and whether or not betting needs to happen. Another observer pattern will allow dealer to know when the table pot is modified so dealer can determine if the betting round is complete or not.

5. Abstract Factory: The abstract factory will bundle Table, Dealer, Player, BettingStructure, and GameType objects when a poker game instance is created.

Class Diagram: