Demo:

https://www.youtube.com/watch?v=UQz2XdVogGk&feature=youtu.be

Java Texas Hold 'Em

Matthew Thompson
Slaton Spangler
Conner Simmering

DEMO

How does it work?

Game Classes:

- Game

- Player

- Opponent

- Deck

- Card

- Ranker

View Classes:

- Main Window

- System Log

Analysis: Sequence Diagram for Game Flow

QUICK background

    Texas Hold 'Em has 3 rounds of betting action

    Betting actions are either: Bet / Check, Fold, and Raise

    We broke actors down into Game, Dealer, and Deck (this
        changed when implemented)

Design

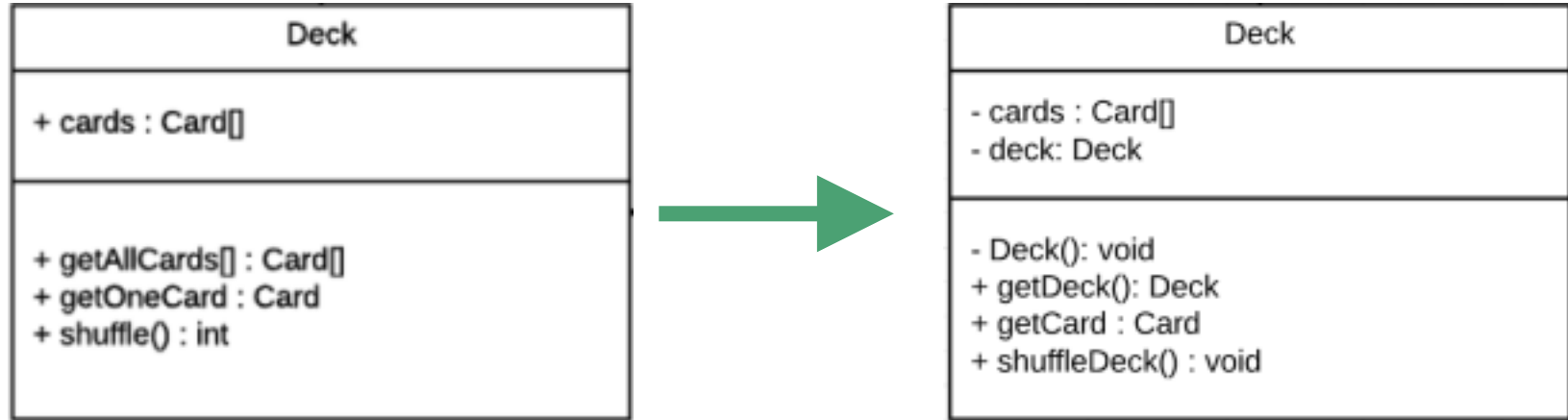Deck class created with singleton design pattern with lazy instantiation

```java
private Deck() {
    random = new Random();
    shuffleDeck();
}
```

```java
public static Deck getDeck() {
    if (deck == null) {
        deck = new Deck();
    }
    return deck;
}
```

Private constructor   Public getter

This was a result of a modification of our original Deck class diagram

Deck
+ cards : Card[]
+ getAllCards[] : Card[] + getOneCard : Card + shuffle() : int

Deck
- cards : Card[] - deck: Deck
- Deck(): void + getDeck(): Deck + getCard : Card + shuffleDeck() : void

Make Deck constructor private and provide method to return (singleton) Deck instance

Instead of a publicly accessible Cards array, provide methods to return and reset (shuffle) the deck