SimpleBlackJack UML

«Abstract»
Player

totalPoints : int
hitChoice : boolean
movesLeft : int

+ hitMe(int points) : void
+ setTotalPoints(int points) : void
+ addTotalPoints(int points) : void
+ getTotalPoints() : int
+ setHitChoice(boolean choice) : void
+ getMovesLeft() : int
+ setMovesLeft(int left) : void

BlackJack

- userPlayer : User
- dealerPlayer : Dealer
- shuffledDeck : Deck
- userHand : ArrayList <Card>
- dealerHand : ArrayList <Card>
- BUST : int
- winner : Player
- whosTurn : Player
- initialDeal : int

+ dealHand() : String
+ hitPlayer() : String
+ stopHitPlayer() : void
+ userScore() : String
+ dealerScore : String
+ userMovesLeft() : int
+ dealerMovesLeft() : int
+ checkBust() : boolean
+ aceIndex(Player player) : int
+ checkWinner() : String
+ checkBlackJack() : String
+ endGame() : void
+ reset() : void

Deck

- numOfDecks : int
- CARDPERDECK : int
- newPack : Card[]
- shuffledDeck : Stack<Card>

+ drawOne() : Card

User

+ stopMe() : void

Dealer

Card

- pointValue : int
- cardName : String

+ getCardName() : String
+ getPointValue(): int
+ setPointValue(int points) : void

Video Demo Link: https://www.youtube.com/watch?v=N65XSVCfYP0&feature=youtu.be
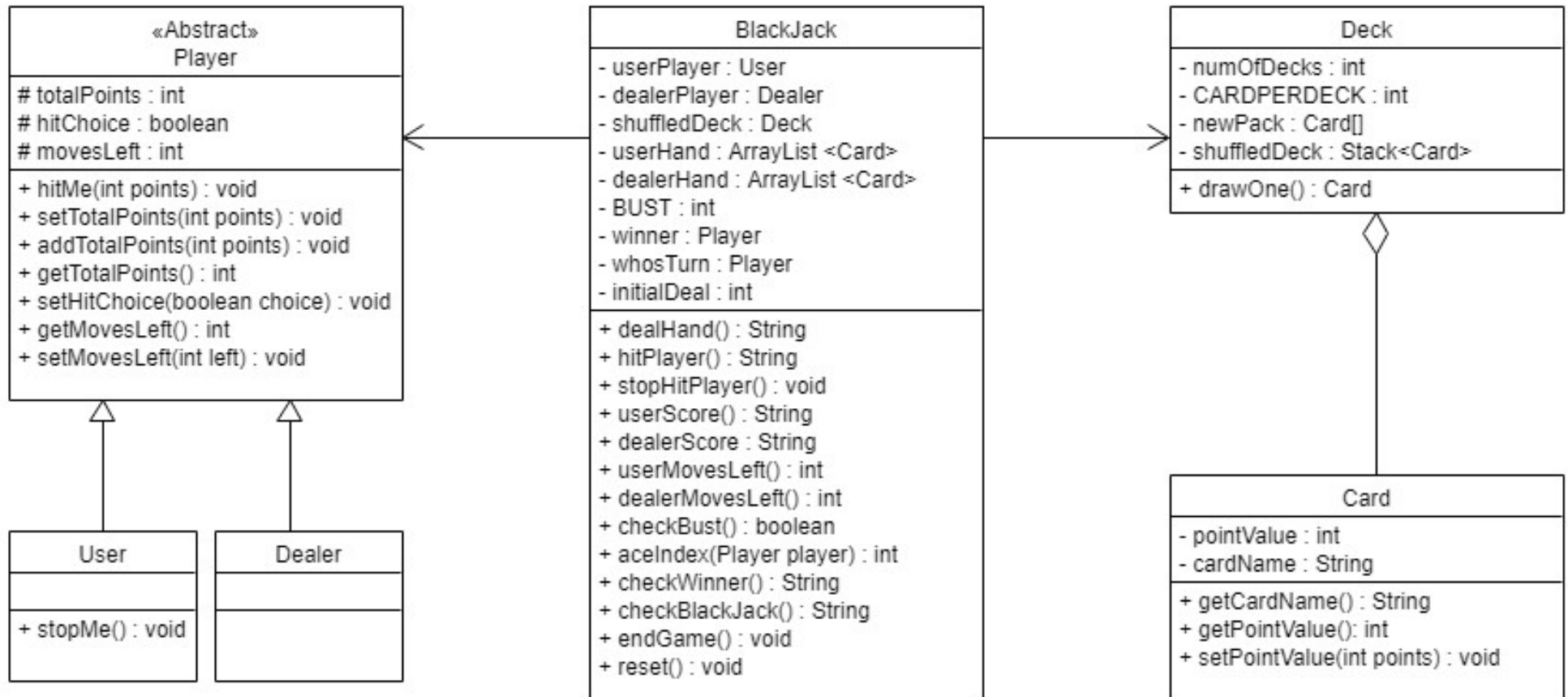
Design:

The classes in the diagram above are the java back end of the SimplyBlackJack app. The BlackJack class represents a model that organizes and manages each and every other class. The User and Dealer are instances of a Player while the Cards are organized and controlled by the Deck.