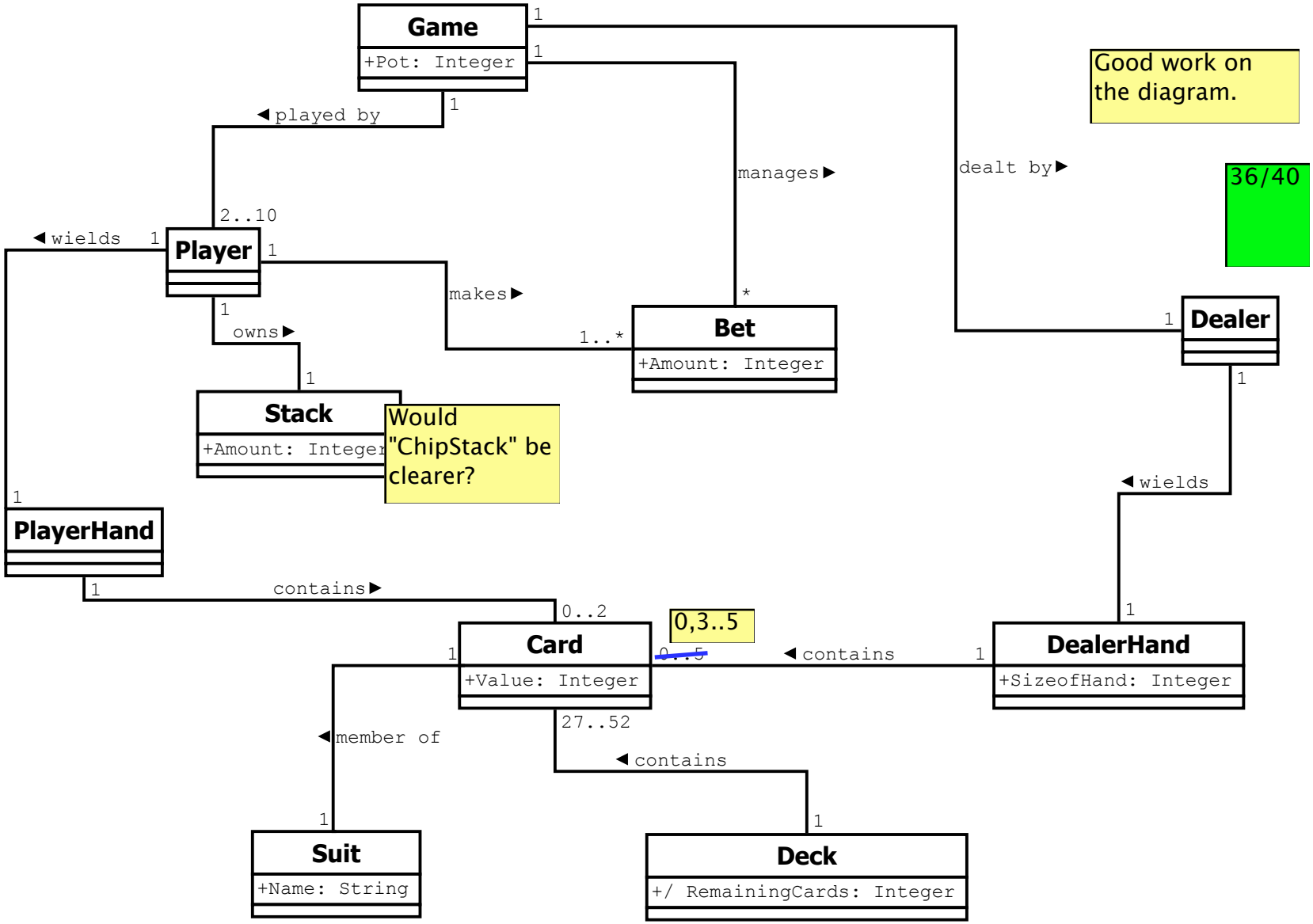
Game
+Pot: Integer
played by
manages
dealt by
Good work on the diagram.
36/40
2..10
wields
Player
makes
owns
Bet
+Amount: Integer
Dealer
Stack
+Amount: Integer
Would "ChipStack" be clearer?
PlayerHand
wields
contains
0..2
0,3..5
Card
+Value: Integer
contains
DealerHand
+SizeofHand: Integer
member of
27..52
contains
Suit
+Name: String
Deck
+/ RemainingCards: Integer

The Game class represents the entire poker game. The Game class has one attribute, Pot, which describes the amount of money currently in play. The Game manages Bets, because there are rules in the game that specify the amounts that a bet can be based on game states. It will manage zero bets at the initialization of the poker game and will manage many bets by the end of the game. A Game is dealt by a Dealer and played by a Player. There is one Dealer dealing a Game, and there are two or more Players playing a Game.

A bit too much info. This is captured unambiguously in the diagram.

A Dealer wields one DealerHand. In Texas Hold 'Em Poker, the DealerHand consists of zero five Cards depending on the round in the game. Therefore, SizeofHand is an attribute of DealerHand. SizeofHand is always zero, three, four, or five based on Texas Hold 'Em play. What round it is in the poker game can be determined by accessing this SizeofHand attribute.

Better to track round directly.

A Card has one attribute: Value. This Value is an integer between two and 14. So, a jack is represented by 11, a queen is represented by a 12, a king is represented by a 13, and an ace is represented by a 14. Each card is a member of Suit. Suit has the attribute Name. The Name is a String that describes the Suit. The String will be "Spade," "Diamond," "Club," and "Heart." The Suits will be implemented using an ENUM to represent the ranking of Suits. A Deck contains all the Cards that are not currently in play. The number of cards in the Deck is an attribute of Deck called RemainingCards. The Deck will always contain between 27 and 52 Cards, because the largest number of Cards that can be in play is 25 (two for each Player and five for the Dealer).

ACK! No. This is supposed to be a domain model. No one calls an Ace of Clubs the 14 of Clubs. You're designing prematurely. (And its a poor OO design to boot.)

Design again.

A Player wields one PlayerHand, which contains zero to two Cards, depending on the round of the poker game. At the beginning of the game, the Player has zero cards. Throughout the rest of the game, the Player will have two cards. A Player also owns one Stack, which has an attribute of Amount. Amount is an integer representing the amount of money with which the Player can use to bet. The Player makes Bets. Since the Player must make a Bet at the beginning of the poker game, the Player must make at least one Bet. The Bets have an attribute, Amount, which represents the amount of money the Player bet.

Don't just describe the diagram. Assume your reader knows the notation. Just describe the conceptual classes and attributes that are unclear, drawing the readers attention to the most critical classes.

I'm concerned that you're missing the whole point of doing a domain model. The words "implemented" and "represents" have no place in a domain model. Let's discuss.