



Exercise: Model BlackJack – DDDE



marcoemrich - last edited 7 days ago [↗](#)

Forked from </s/lVliVj1WUB3pUNjIgnuaLc>

Exercise

model the game BlackJack as a functional domain model using ReasonML types. Don't forget to also model the types of the functions.

BlackJack Rules

Players are each dealt two cards. The value of cards two through ten is their pip value (2 through 10). Face cards (Jack, Queen, and King) are all worth ten. Aces can be worth one or eleven. A hand's value is the sum of the card values. Players are allowed to draw additional cards to improve their hands. A hand with an ace valued as 11 is called "soft", meaning that the hand will not bust by taking an additional card. The value of the ace will become one to prevent the hand from exceeding 21. Otherwise, the hand is called "hard".

Once all the players have completed their hands, it is the dealer's turn. The dealer hand will not be completed if all players have either busted or received blackjacks. The dealer then reveals the hidden card and must hit until the cards total up to 17 points. At 17 points or higher the dealer must stay. (At most tables the dealer also hits on a "soft" 17, i.e. a hand containing an ace and one or more other cards totaling six.) You are betting that you have a better hand than the dealer. The better hand is the hand where the sum of the card values is closer to 21 without exceeding 21. The detailed outcome of the hand follows:

- * If the player exceeds a sum of 21 ("busts"); the player loses, even if the dealer also exceeds 21.
- * If the dealer exceeds 21 ("busts") and the player does not; the player wins.



+ New Sketch



* If both dealer and player receive hands with the same sum, called a "push", no one wins.

Source: Wikipedia

Needed in the Solution

- * draw card(s) (don't forget the effect on the deck)
- * ensure the deck is shuffled
- * calculate scores
- * determine the winner (player or dealer) or draw
- * a way to determine the color of a card for display purposes

Remember

- * there is ****not**** one perfect solution
- * use the terms from the BlackJack domain (ubiquitous language)
- * types should be as business-readable as possible

<> + ✎ + 🗑 -

```
1 type foo;  
   type foo;
```



<> + ✎ + 🗑 -

Exercise 2

make it generic! Extend the game, that it can be played with an European deck of cards using numbers from 7 to 10.

<> + ✎ + 🗑 -

```
2
```

<> + ✎ + 🗑 -