BLACKJACK TERMINAL APP

•••

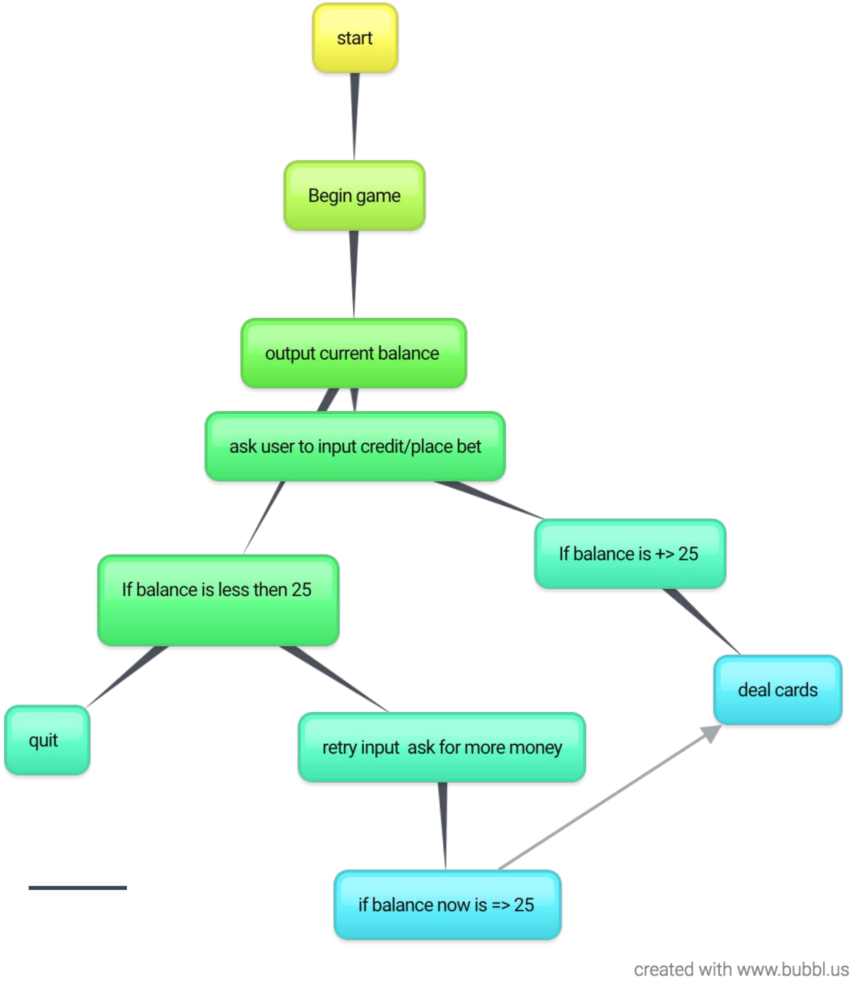
By Mitchell & Glenn

RULES

- Application is one player only, you are the player, playing against the dealer
- Aces will be counted as 11
- Face cards will be counted as 10 ie. King, Queen, Jack
- The value of the hand is the sum of the point values of the individual cards
- After the player has placed initial bet, the dealer will shuffle the deck and give 2 cards to the player and 2 cards to himself.
- Player has 2 options:
- "Hit" - Player draws another card (and more if he wishes). If this card causes the player's total points to exceed 21 then he loses.
- "Stay" - Player is happy with his cards, and move to dealer's turn.
- The dealer has the same options a previously stated above
- If the dealer goes over 21 points, then player will win
- If the dealer doesn't bust,then the point total that is closest to 21 between the player and dealer

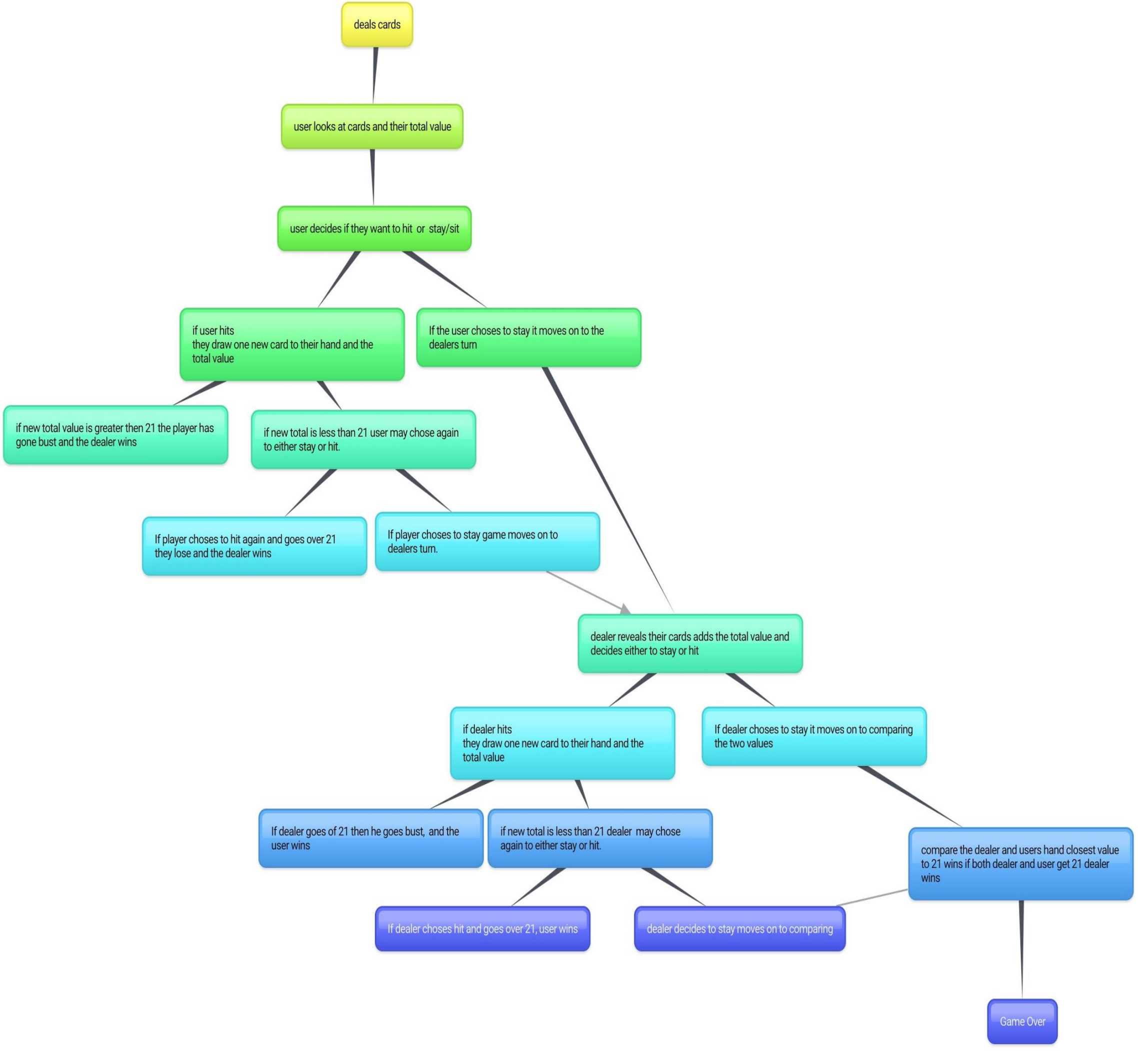
User Flow pt 1

- Welcome the user
- Ask for minimum bet of $25 to start the game
- After the the user submits a bet greater than $25, we deal them their first two cards


start
Begin game
output current balance
ask user to input credit/place bet
If balance is less then 25
If balance is +> 25
quit
retry input ask for more money
deal cards
if balance now is => 25
created with www.bubbl.us

User Flow pt 2

- Deal the cards to the user
- Ask the user if they want to hit or stay
- If they chose hit and the total card value goes over 21 dealer wins
- If they hit and card value stays under 21 repeat hit or stay question
- If stay move on to dealers turn
- Dealer gets asked hit or stay and the same rules as previously stated apply
- If both dealer and player dont bust and chose to sit compare the cards
- Whoever has the closest value to 21 wins however if both player and dealer both had 21 dealer wins



deals cards

user looks at cards and their total value

user decides if they want to hit or stay/sit

if user hits
they draw one new card to their hand and the total value

If the user choses to stay it moves on to the dealers turn

if new total value is greater then 21 the player has gone bust and the dealer wins

if new total is less than 21 user may chose again to either stay or hit.

If player choses to hit again and goes over 21 they lose and the dealer wins

If player choses to stay game moves on to dealers turn.

dealer reveals their cards adds the total value and decides either to stay or hit

if dealer hits
they draw one new card to their hand and the total value

If dealer choses to stay it moves on to comparing the two values

If dealer goes of 21 then he goes bust, and the user wins

if new total is less than 21 dealer may chose again to either stay or hit.

compare the dealer and users hand closest value to 21 wins if both dealer and user get 21 dealer wins

If dealer choses hit and goes over 21, user wins

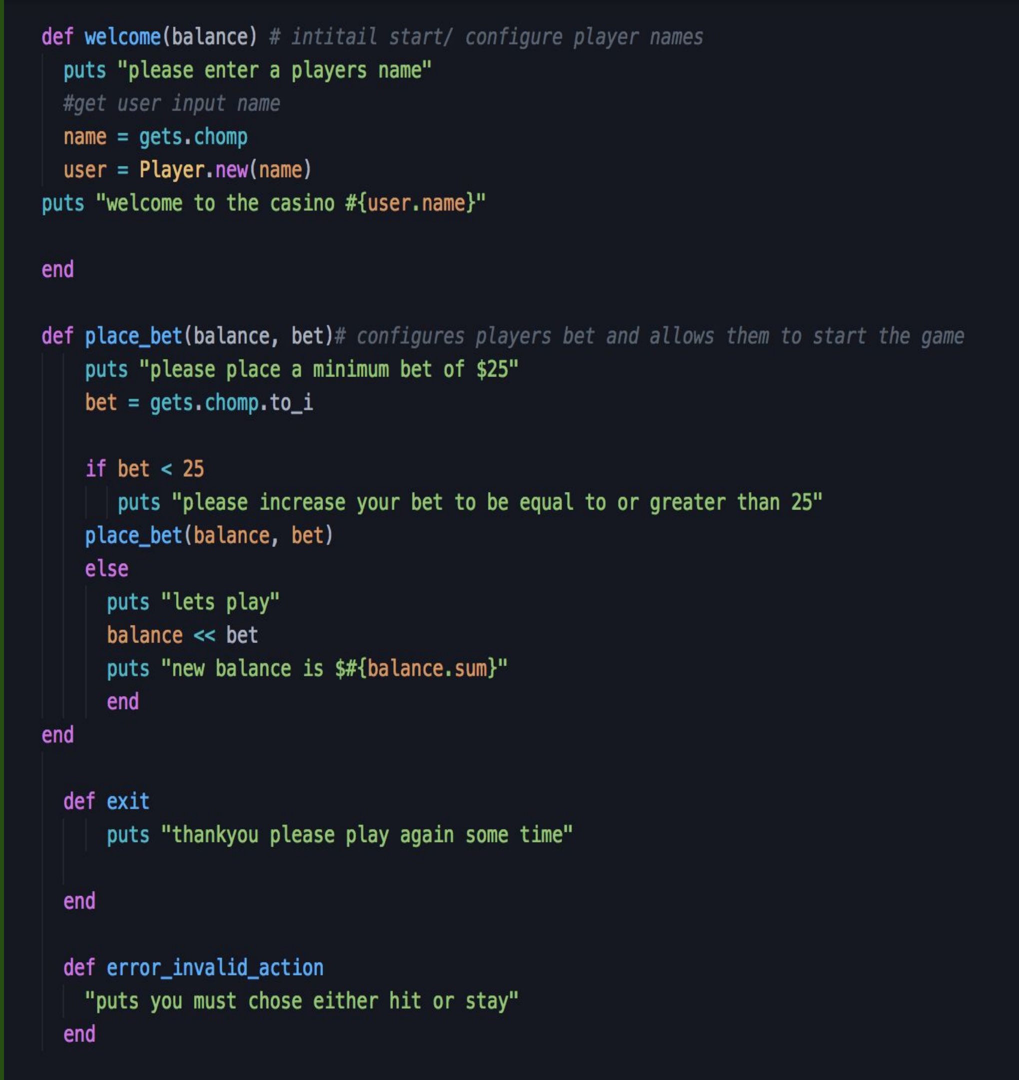
dealer decides to stay moves on to comparing

Game Over

created with www.bubbl.us

Build Process

For the build process of our app we followed the user workflow to make sure we forgot no crucial aspects of our app an example of this is displayed to right of the slide which depicts our welcome and place bets method



```
def welcome(balance) # intitail start/ configure player names
 puts "please enter a players name"
 #get user input name
 name = gets.chomp
 user = Player.new(name)
puts "welcome to the casino #{user.name}"

end

def place_bet(balance, bet)# configures players bet and allows them to start the game
  puts "please place a minimum bet of $25"
  bet = gets.chomp.to_i

  if bet < 25
   puts "please increase your bet to be equal to or greater than 25"
  place_bet(balance, bet)
  else
   puts "lets play"
   balance << bet
   puts "new balance is $#{balance.sum}"
   end
end

 def exit
   puts "thankyou please play again some time"

 end

 def error_invalid_action
  "puts you must chose either hit or stay"
 end
```

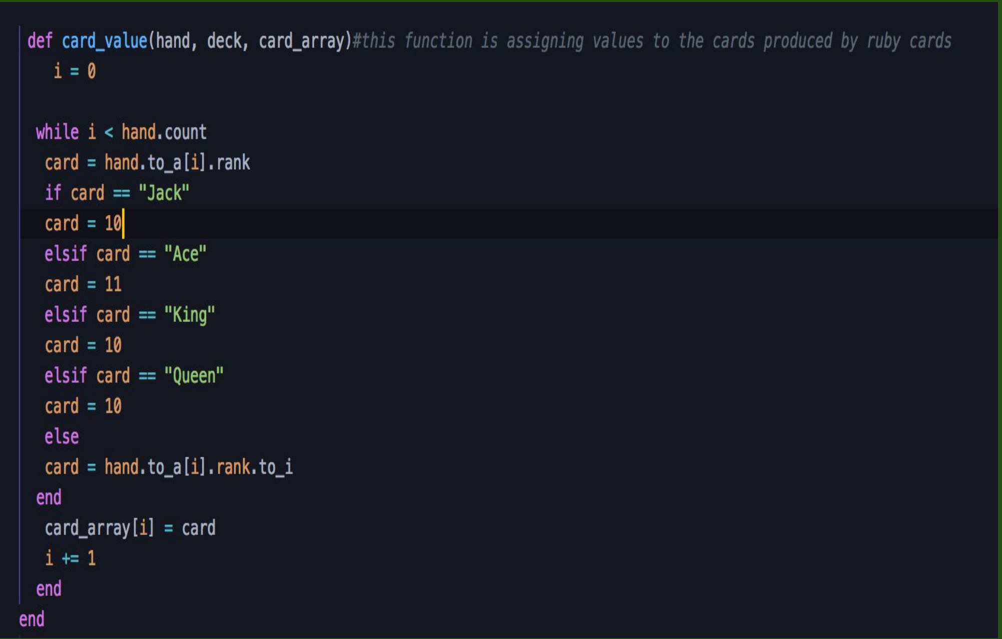
Ethical Issues Regarding Our App

- The two main ethical Issues in relation to our app is the encouragement of gambling and Underage kids being influenced by gambling games such a this one.
- However our app is pretty simple and the chances of it influencing a gambling addiction later on in the user life is slim to none

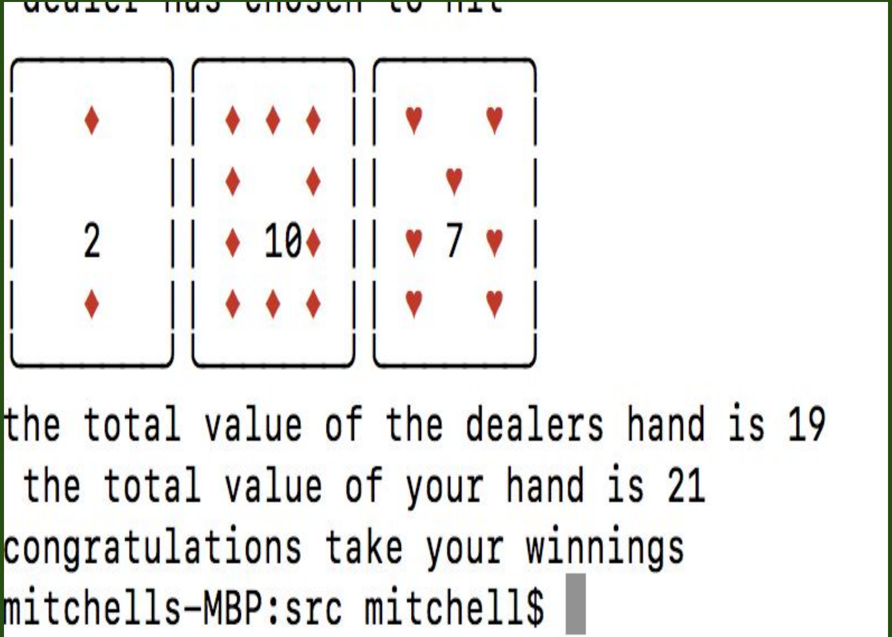
CHALLENGES

- We had some issues implementing RubyCards gem into our application, in particular giving each card a value, and then summing these cards together to give a total value.
- We addressed the issue of RubyCards gem having no value by creating a function that assigned values to the cards.
- However after fixing the issue of the cards having no value we, ran into the issue that the ace cards can either have a value 11 or 1 at the user's discretion, for the sake of completing the app we decided to assign ace to the value of 11

Challenges

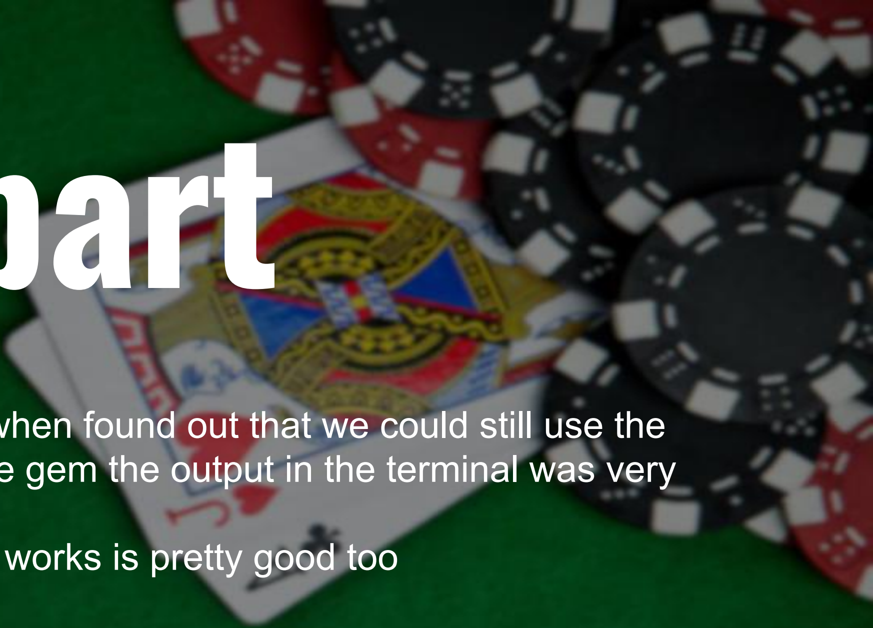


```
 def card_value(hand, deck, card_array)#this function is assigning values to the cards produced by ruby cards
    i = 0

  while i < hand.count
   card = hand.to_a[i].rank
   if card == "Jack"
   card = 10
   elsif card == "Ace"
   card = 11
   elsif card == "King"
   card = 10
   elsif card == "Queen"
   card = 10
   else
   card = hand.to_a[i].rank.to_i
  end
   card_array[i] = card
   i += 1
  end
end
```

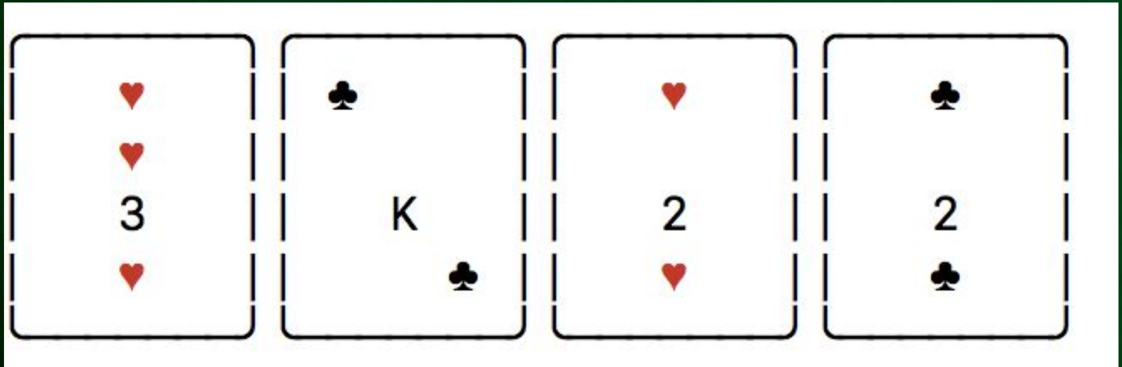


```
dealer has chosen to hit

┌─────────┐┌─────────┐┌─────────┐
│    ♦    ││ ♦  ♦  ♦ ││ ♥     ♥ │
│         ││ ♦     ♦ ││    ♥    │
│    2    ││ ♦  10♦  ││ ♥  7  ♥ │
│    ♦    ││ ♦  ♦  ♦ ││ ♥     ♥ │
└─────────┘└─────────┘└─────────┘

the total value of the dealers hand is 19
 the total value of your hand is 21
congratulations take your winnings
mitchells-MBP:src mitchell$ 
```

Favourite part



- The best part of this project was definitely when found out that we could still use the RubyCards gem, this is because without the gem the output in the terminal was very bland.
- And of course the fact that the app actually works is pretty good too



```
    ♥        ♣                 ♥           ♣

    ♥

    3             K            2           2

    ♥                  ♣       ♥           ♣
```

IMPROVEMENTS

- Implement the ability to have multiple players, even possibly make the dealer a playable character to make the game more challenging.
- Store user balances and their winnings so that they can continue where they left off when they play again
- Maybe even an age verifier to address the ethical issues we spoke about earlier
- Include some prompts at the start of the game that makes the player aware of the ethical issues of gambling and specify that this app is just a fun game and in no way is it a representation of these issues
- Make it so the user is able to decide if they want the ace in their hand to be either equal to 1 or 11
- Use the colorize gem to make certain prompts stand out more

APP DEMO

RESOURCES

- Project management tools e.g. Slack, Trello
- RubyCards Gem
- Official Ruby Docs
- Stackoverflow
- Stackexchange
- bubbl.us

QUESTIONS