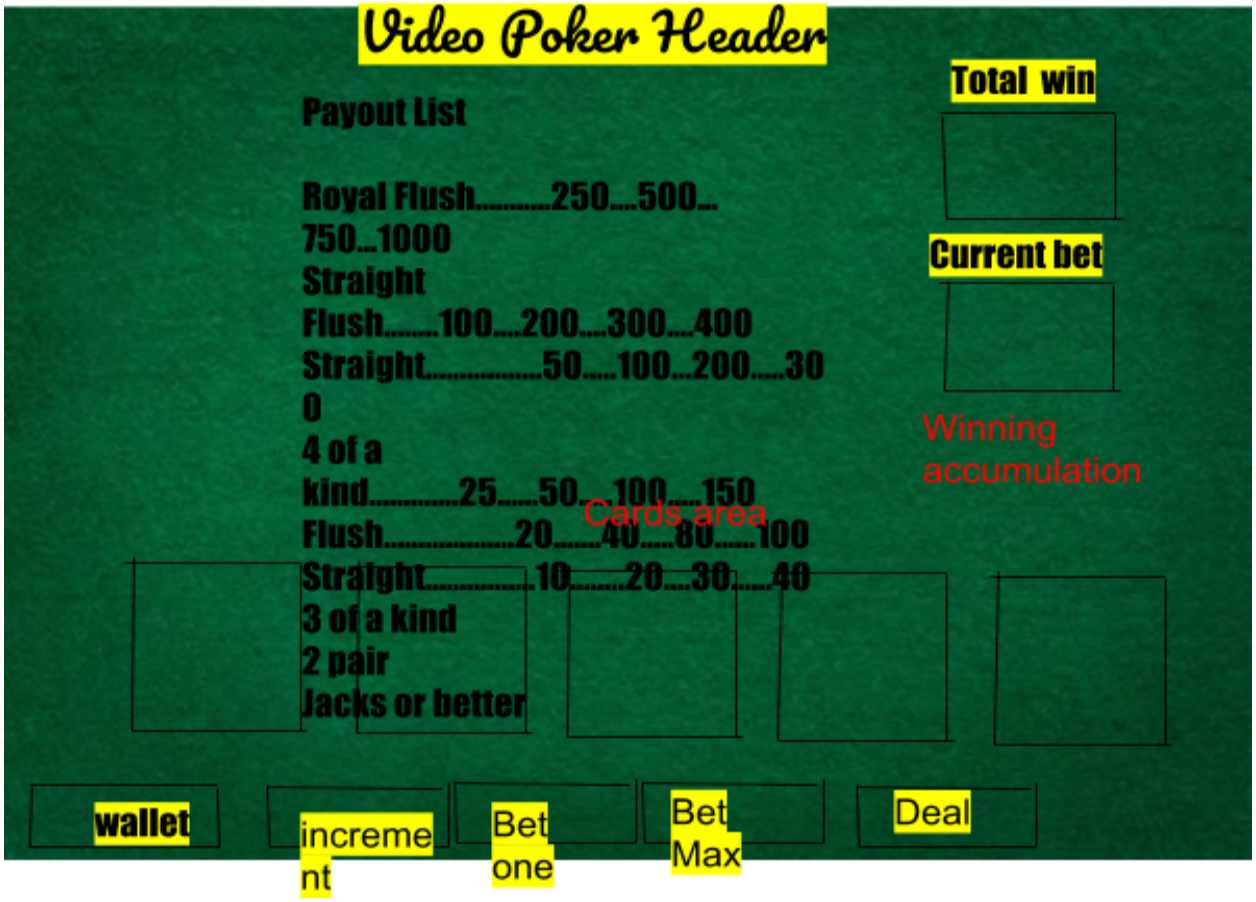
Video Poker

PsuedoCode

- Player starts game
- Player is shown screen with empty deck of 5 spaces
- Before a deck is randomized, player needs to place a bet
    - Player can place a bet of either 5, 10, 50, 100 denominations
    - These denominations represent a dollar amount
    - The incentive to place a bet is to hit a multiplier if a successful ‘suit’ is achieved
- Once bet is complete, player presses the deal button
- The program will randomize 5 cards from a 52 card deck
    - The remaining cards, minus the cards that were just dealt must be kept in memory
- The player makes a decision to keep any number of cards or give up any number of cards
    - If the player has kept all the cards, the program needs to check if a suit is matched or high card (jacks, queens, kings, aces)
        - If a suit is achieved, for example, a ‘straight’, the player will be paid out, times the multiplier for that suit. For example, a straight suit is a 50x multiplier.
        - Game will present a winner's animation and reset back to the start of the game.
            - If no suit is achieved, players money is kept, game restarts
    - If the player has eliminated cards, which could either be 1 card or up to all 5 cards, the program will retrieve another set of cards, randomized from the remaining cards left to replace the chosen cards, but not before:
        - Additional bets can be made,
        - New cards are dealt from remaining cards from the pack
        - Player will then receive new cards.
        - Once new cards are received, programming will check if there is a suit or high card, and pay out accordingly.
            - If no suit or high card is achieved, player’s money is deducted from wallet and game restarts.

Wireframe



Video poker architecture:

- **Game states:**
  For example:
  var gamestates {
  Uninitialized: 0,
  Firsthand: 1
  Etc.

Get card resources from git. General assembly

Create a deck of cards with loops

- Regular deck contains 52 cards
- 4 suits
- And 13 ranks from 2 to ace
- Create an an empty array for the deck
- First tell the computer all the ranks and suits that exist
    - variables suits and ranks
    - You create an array, for example H C D S, using single quotes for text based strings (hearts clubs, diamonds spades)
    - Than ranks 2, 3, 4 etc, to T (ten), jacks queen k A
- We create a for a loop to iterate over these arrays
    - We write 3 statements: first statement will be the suit counter = 0, which is its starting value
    - Second statement indicates when how many loops for the suits, for example: <4
    - The the third statement increments through the suit counter (++)
        - For practice: Now in the for statement, we want to count and display all the suits, eg, HCDS
        - Do this with the ranks
    - Now we want to create a nested for loop (learn more about this)
        - This should go through all ranks and suits
        - For practice, we can concatenate these two and console.log
    - Push the deck on to the empty array.

Shuffle the deck of cards: