**Black jack** : save game(keeping track of player chips), resume, AI?, SVG

-**menu** 1. New game

-default sum given to player

-randomize card 1deck of cards

-dealing the cards- one for dealer, one for player x2

-rules for the game are enforced

-elimination of cards dealt from the deck

-end game if player has 0$, or exit(with option to save progress(in game menu))

2. Load game

3. Exit

-**in game menu**: 1. Fold(lose the round)

2. hit(get another card)

3. How much money to bet

4. Save and quit

5. Quit without saving

**Rule:**

the house must keep asking for cards if total number is under 16, stand at 17. Players are able to keep asking for cards as long as sum isn’t over 21. If the player bust(eg. 22), the round is over. The chips are forfeited to the house and total chips won from player increases. If the player has 0$ in chips, the game is over (display good game, returns the main menu after). The deck is reshuffled when 4/5 of the cards have been dealt.

**Approach:** create a stack using linked list as the **deck**. Must be randomized prior to dealing cards(fisher-yate shuffle vs random\_shuffle vs rand()?)(assign suits based on card # 1-52, eg. increments of 4). A single deck contains 52 cards.

**Classes:**

player(player info): name, chip amount, hand(cards dealt)

blackjack(game rules, deck): number of cards in deck, what cards have been dealt array, total chips won from player.

Object card: number, suite

Create an array of 52 card objects, then shuffle them. And push it into a list?

**-functions:**

-check who wins

-check for bust

-check for 21(if one ace is revealed, the player has the option of buying insurance. Check if the house has 21. If so, then the player loses. If the player also has 21, it is a draw. Nobody wins)

-starting bet amount

-reveal card

-show rules

-startgamemenu-> New game: call the newgame function

->Load game: calls load game function

->exit: exit loop/program

-asking for player’s name function (new game’s constructor)

-giving a default chip amount to player (initialize player’s chip amount) (new game’s constructor)

-dealing cards function: deal one card to player, one to house, another one to player, and last one to the house. One card dealt to the house must not be visible.

-draw the card function: (optional)\_\_\_\_\_\_\_\_\_\_

|                    |

 |     king       |

 |                   |

 | \_\_\_\_\_\_H \_|

-save and quit function: save the state of the current round. Write to file player’s name, chip amount. Save the deck’s state (cards in the deck and cards already dealt)

-load function : initialize the player’s name and chip amount , redeal the cards