

Final Project Proposal

Casino

For our final project, we will be creating a casino where you start off with a set amount of money. With that money you can play games which costs money to play. You can either lose all your money you used or gain some money. When you are left with \$0.00, the game is over and you've gone bankrupt.

Our project employs Super/sub classes (in blackjack where player and dealer are both subclasses of 'blackjackUser', but with different methods, as the rules differ for a player and a dealer), array sorting and manipulation (in connect four with the arrangement of the pieces, players play on an `int[][]` grid), dealing with keyboard input (How the player interacts with the 'casino'). Implements interface, as blackjack and hiLow use many of the same methods to figure out who won.

(start with connect four and make our way down the list, may not get everything done)

Goal: connect four, wheel of fortune, and blackjack

Minigames include:

1. Connect Four

- a. Play against a computer as you strategically place your pieces, attempting to connect four of your pieces in any straight manner.
- b. There will be a board set up with labeled positions.
- c. User chooses where to place piece of choice.
- d. Once 4 in a row is achieved, user wins.
- e. Computer does search for 3 in a row, if not then 2 in a row to try to complete a 4 in a row.
- f. Players play on an `int[][]` grid, where 0 marks open spaces, 1 marks player's pieces, 8 marks computer's pieces
- g. After every move, a new grid with updated moves is flushed to the terminal

2. Black Jack

- a. The player will play against 'the house'
- b. Limited to one hand at a time, no splitting
- c. You are dealt your cards, which are displayed in the terminal. You also see one of the house's cards.
- d. You have the option to hit or stay, attempting to reach 21 without going over
- e. After you have finished your moves, the computer deal goes

- f. Has a 'deck' array, uses that to simulate playing cards. Once a card is dealt, removed that particular card from the array. The array does not distinguish between suites, as it doesn't matter in blackjack
 - g. Includes object arrays, type casting, strings, and ints
- 3. Hi - Low card game
 - a. You bet a certain amount of money, and then are given a card at random from a deck
 - b. The computer is also given a card
 - c. Whoever has the higher card wins, and the money won or lost is proportional to the difference of value in the cards
 - d. Will implement interfaces, as many of the same methods are used by blackjack