

Group 10 - Blackjack Card Counter

Nick Johnson
Clayton McPhail
Jessica Melville
Victoria Fleming
Sam Cvikota
Zachary Walberg



Project Overview

- Create working Blackjack game with emphasis on learning how to card count
- Give players an edge over the house by learning the simple method of High-Lo card counting to make better decisions
- Better decisions = losing less money in the long run



Problem Statement

- Sharpening people's mental math abilities
 - Including the mental math practice in a the form of a game increases the chances of retaining skills
- Combat boredom
 - Stimulation of the mind enhances mental health and helps fight dementia and alzheimers

Blackjack

Goal:

- Beat the dealer
 - Add cards to get hand close or equal to a total of 21
 - Automatically lose if you go over 21

Gameplay:

- Dealer and player starts with 2 cards each
- “Hit”: Dealt another card
- “Stand”: Stay at current total and dealer plays
 - Ace = 1 or 11
 - 2 - 10 = face value
 - Face cards = 10



High-Lo Card Counting

Very Simple Method of Card Counting.

- Gives player a better way to estimate the next decision to make
- Low value cards benefit Player
- High value benefits Dealer



Strategy - Getting Started

- First step was to learn the game of blackjack
- Choose our method of card counting
 - Very simple hi-lo count (below)
- Define the cards and to tracking
- Draw cards
 - Create a draw function using a random number generator

Card Strategy	2	3	4	5	6	7	8	9	10, J, Q, K	A
Hi-Lo	+1	+1	+1	+1	+1	0	0	0	-1	-1

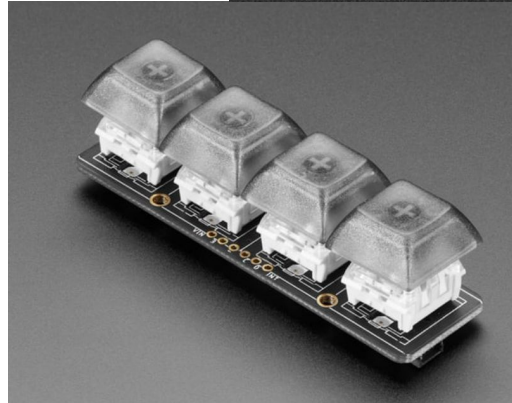
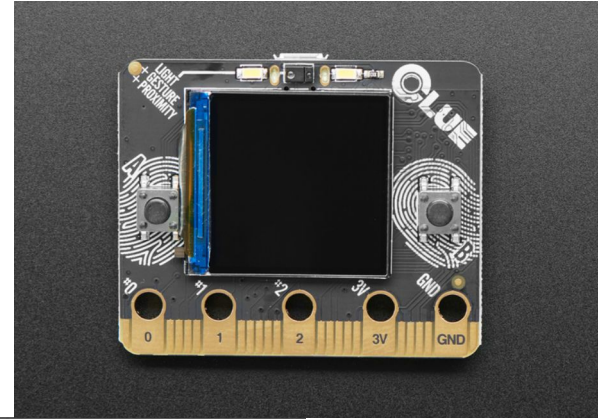
Image Credit: Wikipedia - Card counting
https://en.wikipedia.org/wiki/Card_counting

Getting a Working Blackjack Game

- We had to figure out how to track cards dealt
 - Also had to make sure cards were not dealt too many times
- Print correct card values and suits
- Design logic and track totals
- Create conditions to exit game

Strategy - Hardware

- How to use board, examples
- Write feature functions
 - Hard-coded variables
- Incorporate game code



Solution - Software

- Card generation
 - Read in matrix from CSV file
 - Ended up hardcoding onto arduino
- Draw function
 - Use random number generator
 - Track and use random value to pull cards

```
upper = 51;
lower = 0;

srand(time(NULL));

for(i = 0; i < count; i++){
    draw = (rand() % (upper - lower + 1)) + lower;
    //printf(" %d",draw);
}
//printf("\n");

return draw;
```

Solution - Software

- Print functions
 - Face card printing (middle)
 - Card value printing (left)
- Card Counting (right)

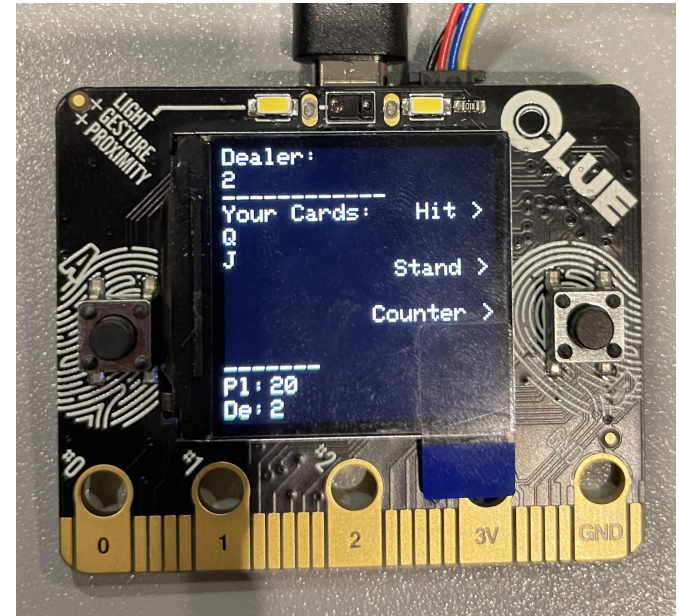
```
printf("Dealer: ");
for(i = 0; i < dealer_cards; i++){
    card = cards[dealer_hand[i]][1];
    if(dealer_cards == 2 && i == 0){
        printf("? ");
    }
    else{
        if(card == 1 || card > 10){
            high_val_print(card);
            suit_print(cards[dealer_hand[i]][0]);
        }
        else{
            printf("%d ", cards[dealer_hand[i]][1]);
            suit_print(cards[dealer_hand[i]][0]);
        }
    }
}
if(dealer_cards == 2 && player_done == false){
    printf("\nDealer Total: %d\n\n", cards[dealer_hand[1]][2]);
}
else{
    printf("\nDealer Total: %d\n\n", dealer_total);
}
```

```
switch (face){
    case 1: card_face[0] = 'A'; break;
    case 11: card_face[0] = 'J'; break;
    case 12: card_face[0] = 'Q'; break;
    case 13: card_face[0] = 'K'; break;
    default: card_face[0] = 'N'; break;
}
printf("%s ", card_face);
```

```
if (card > 1 && card < 7)
{
    val= 1;
}
else if (card >= 7 && card < 10)
{
    val= 0;
}
else if (card == 10 || card == 1)
{
    val= -1;
}
else
//debug
    printf("Invalid Range (cardval check)\n");
return val;
```

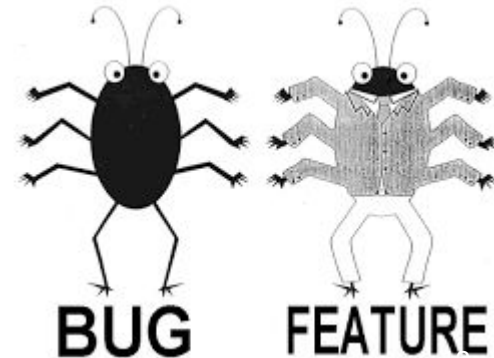
Solution - Hardware

- Clue Board
 - Arcada Library
 - Display format and text
 - Board buttons
 - Right displays counter, left clears
- NeoKey
 - Relied on Adafruit SeeSaw Library for examples
 - Keys will light when the neokey is ready
 - Keys relate to hit or stand in the game
 - Keys will light up green when a hand is won and red when a game is lost
- Integration
 - Replaced terminal functions with feature functions



Status - Software

- C code includes the basic steps to play BlackJack
- Minor bugs/improvements can be made to the game-logic
 - Bugs
 - The code's logic for Aces is hard to balance for different scenarios
 - ...
 - Improvements
 - Add more advanced logic for BlackJack game
 - Split function
 - Card counter
 - More players or decks



Status - Hardware

- Integrated the fully functioning C code into the Arduino syntax
- Game operates on the CLUE Board
 - Successfully
 - All the functions work in Arduino syntax
 - Integrated the NeoKey Buttons into the code and CLUE board
 - Both buttons on board and 2 on NeoKey are used
 - Unsuccessfully
 - The hands chosen will repeat if the board is reset via the button on the back of the board

Demonstration



Questions?

References

[Arduino Docs | Adafruit NeoKey 1x4 QT I2C Breakout | Adafruit Learning System](#)

[Overview | Introducing Adafruit CLUE | Adafruit Learning System](#)

Adafruit Arcada CLUE Full Board Test Example

BETMGM. (2022). *Bird's eye view of Blackjack table*. BETMGM. Retrieved 2022, from <https://casino.betmgm.com/en/blog/what-are-the-different-types-of-blackjack-tables/>.