Arduino Blackjack

This project utilizes user input to simulate a game of blackjack. It uses a potentiometer to allow the user to adjust his or her bet, an LCD to display balance and cards dealt, and it continues gameplay until the balance is zero.

Braeden Benedict, Brooks Meadowcroft, Samantha Scaglione

1. Executive Summary

The “Arduino Blackjack” project is a blackjack playing toy that was created using an Arduino, a potentiometer, an LCD display, and a computer keyboard and monitor for user interaction. The user, initialized at the start of a game with 100 dollars, is able to select a betting amount using the potentiometer then confirm the betting amount displayed on the LCD screen by pressing “enter” on the computer keyboard. Once the cards are dealt and displayed on the LCD, the player is able to choose whether to 'hit' or 'stand' by typing “h” or “s” using the keyboard attached to the monitor screen. The LCD is used to display both the player’s cards, one of the dealer’s cards while the round is being played, and the current betting amount. As in regular blackjack, the player is playing each round against the dealer, who plays with set rules. These set rules are that the dealer must hit if his hand is worth less than 17. Gameplay will continue until the player runs out of money or chooses to exit. This program will be coded on an Arduino microcontroller using the Arduino C/C++ programming language.

When the group first decided to build a blackjack game using an Arduino, their intention was to use buttons as inputs through which the user would be able to choose whether to hit or stand. However, the group ran into issues and decided to implement our “contingency plan” of prompting the user for input through the desktop or laptop that the Arduino is hooked up to. Before deciding to proceed with the contingency plan, the group double and triple checked the wiring of the hardware. The group also used a digital multimeter to ensure that the buttons were not in some way broken or shorted. Furthermore, the group attempted to switch out the buttons for other buttons and even attempted to wire them to a different breadboard. All of these attempts were to no avail, so the group settled on having the player interface with the game through the computer keyboard.

2. Introduction

The goal of this project was to create a blackjack playing toy through the use of an Arduino, a potentiometer, an LCD display, and a computer keyboard and monitor for user interaction. The group was interested in using the blackjack simulator to entertain and engage the player through the use of the potentiometer to adjust his or her bet and the LCD display of cards dealt, bet, and balance. Although the group did not originally plan to use the computer keyboard to get input from the player, it ran into problems in attempting to use the buttons as input as will be discussed in *Section 3.4: User Input*. After adjusting the original plan, the group decided to have the player type "h" in order to "hit" and be dealt another card or "s" to "stand" with the cards he or she was already dealt.

The Arduino language is a cross between C and C++, so, although the group was experienced with C from in-class activities, the group found itself constantly checking the Arduino website for differences in syntax. The group found that one major, and helpful, difference between the Arduino language and C is that the Arduino has a built in random number generating function that allows a minimum and maximum value to be set without going through the process of setting up conditional statements to ensure that parameters are met. In addition to consulting the Arduino website to ensure the correct syntax was used, the group also turned to the website to get an idea of how hardware should be wired to the breadboard and the Arduino itself.

The LCD wiring was based heavily on the diagram shown below in Figure 1. While wiring the LCD, the group opted to preserve room on the breadboard and limit the confusion that having two potentiometers. The third LCD pin was originally connected to a potentiometer to allow the user to vary the brightness of the LCD screen. The group decided that it would be

simpler to keep the screen on the brightest setting and therefore directly ground the third LCD pin.

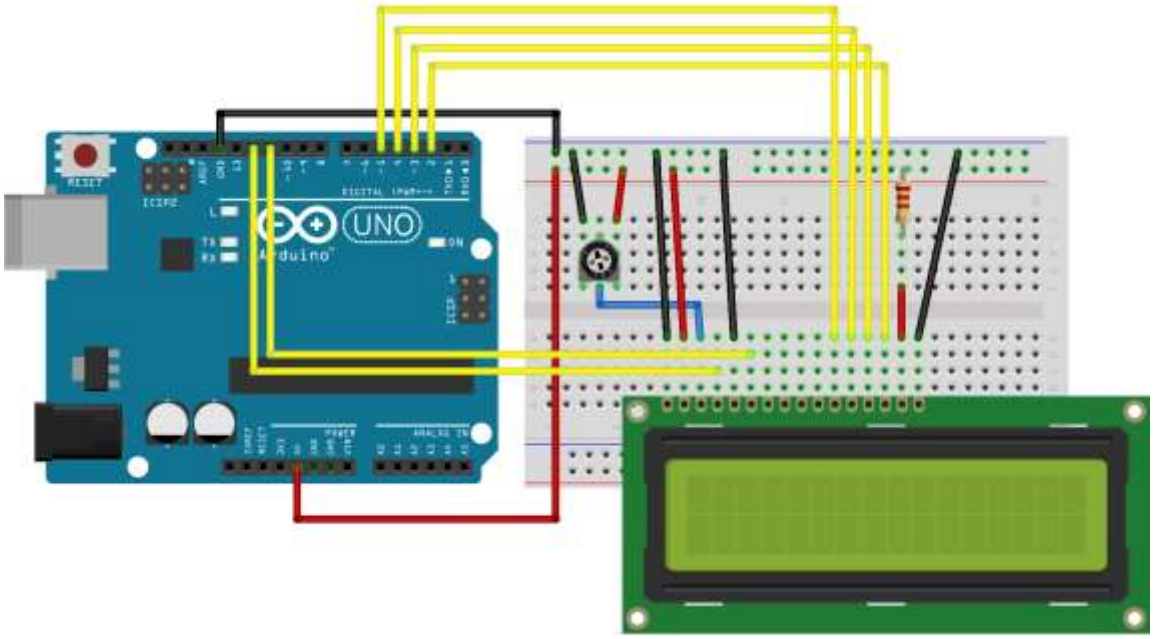

Figure 1: “Hello World!” Schematic (Source: www.arduino.cc)

3. Methodology

3.1 Overview and Main Control Flow

The “deck” of 52 cards is stored in an array of single characters (A,2,3,4,5,6,7,8,9,T,J,Q,K) ordered by number, then by suit. Its character is used to determine its value and its position within the array (i<13, 12<i<26, etc) is used to determine its suit. In the event that the player draws an ace, it is automatically counted as 11 unless the player exceeds 21, as per the rules of blackjack. The player is given a starting balance of $100. Appropriate delays are included throughout the program to simulate cards being dealt as in an actual game of blackjack.

After the player selects a betting amount using the potentiometer, both the player and the dealer are initially “dealt” two cards using a deal function. Instead of actually shuffling the order of the cards, the function chooses a random number from 0 to 51 when dealing cards to simulate the deck having been shuffled. The indexed value is used then replaced by the null character after the card has been dealt. The function runs such that it generates a new random number until it finds an array entry that is not the null value.

The LCD displays one of the dealer's cards and both of the player’s cards. In place of the dealer’s second card it displays “XX.” If the player’s cards sum to 21, the round is over and 1.5 times the bet is added to the player’s balance. If both the dealer and player’s cards are 21, the round is over and there is no change in balance. If only the dealer’s cards are 21, the round is over and the bet is subtracted from the balance. If neither has 21 at this point, the player is prompted to either “hit” or “stand.” If he chooses to hit, the deal function is used to deal the player an additional card. If the player’s cards now sum to over 21, the bet is subtracted from the balance. If not, the player will be once again given the option to hit or stand. This process will repeat.

Once the player chooses to stand, both the dealer's cards are shown face up. If the sum is less than 17, the dealer will be dealt an additional card. If the dealer's cards sum to over 21, the player wins and the value of the bet is added to balance. If the dealer's sum is less than 17, this process will repeat. If the sum is 17 or greater, the dealer's sum will be compared to the player's sum. If the player's sum is greater, the bet is added to his balance. If the dealer's sum is greater, the bet is subtracted from the balance. If they are equal, there is no change in balance. The round is now over.

At the end of each round, the outcome of the round is displayed- "won round" or "lost round" along with the amount of the player's money left. After each round is "won" or "lost," the deck of cards array is reinitialized (null characters will be replaced by their original characters). Gameplay continues until the player runs out of money or chooses to exit by pressing the reset button on the Arduino.

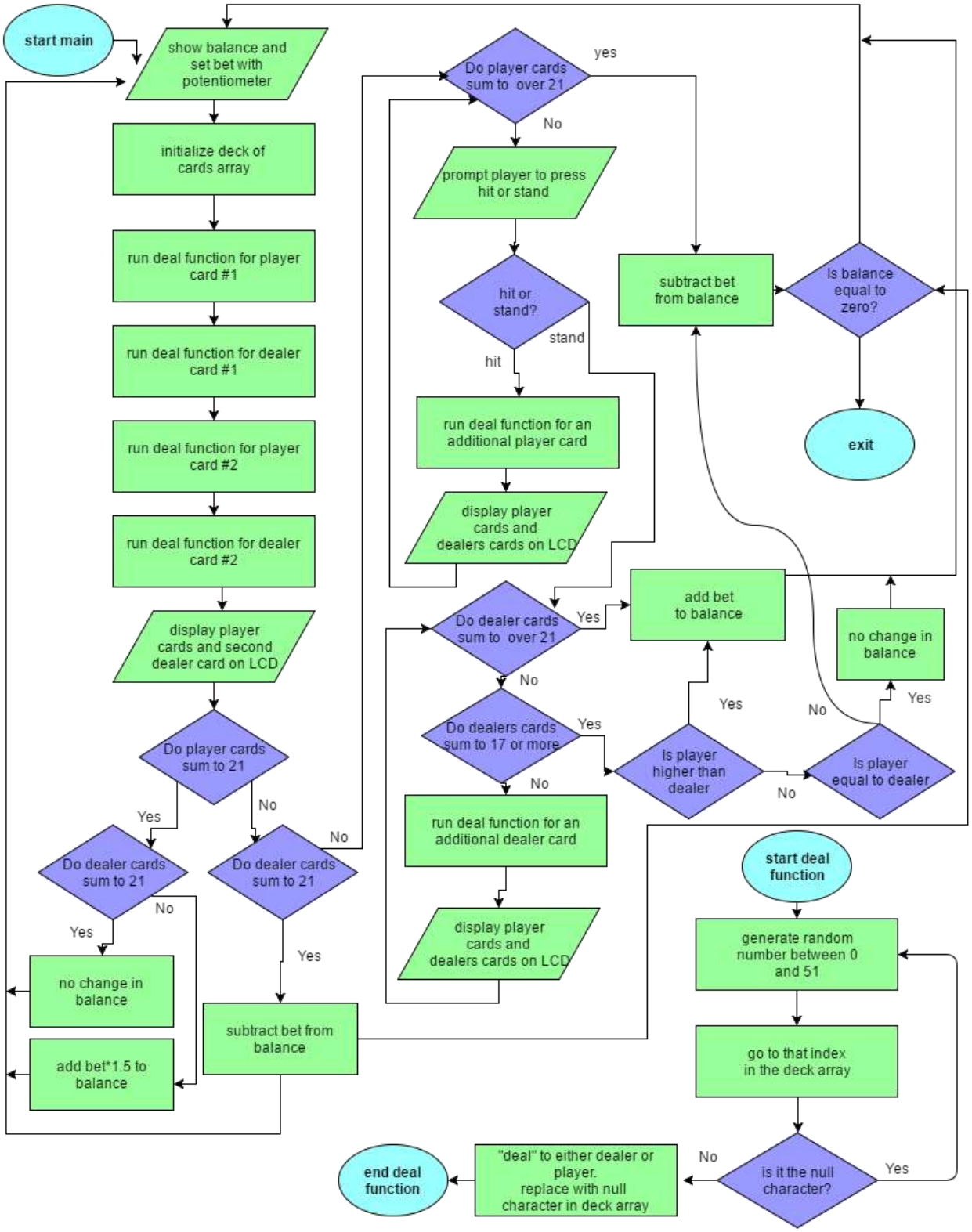

Figure 2: Program Flow Chart

3.2 Hardware/LCD Customization

As shown above in Figure 1, the wiring for the LCD was somewhat convoluted. The team first wired the LCD exactly as shown in the diagram. After recognizing that the potentiometer pictured was only in place to allow the user to adjust the brightness of the LCD screen, the team decided that, in order to minimize confusion with the other potentiometer that would be added to control the bet, the team would set the LCD to always be at the brightest setting by connecting it directly to ground rather than through the potentiometer. Furthermore, after setting up the wiring of the LCD, the team realized that the potentiometer and the buttons that the group originally planned on using may not fit on the Arduino's relatively small breadboard. The team rearranged the LCD placement and wiring in order to accommodate more room for the other features of the blackjack game. When thinking forward to the display that would be pictured on the LCD, the team thought using a "h" to represent hearts would yield a lackluster experience for a potential player. To avoid this, the team researched how to create custom characters to display on an Arduino LCD. By defining byte by byte (instituting a 1 for a lit space and a 0 for a dull one) which pieces of the array in the LCD display to light up, the team defined a different custom character for each suit. They filled in the traditionally red suits and left only the outline for the traditionally black suits. The mocked up symbols are displayed in Figure 3 below and the finished symbols on the screen of the LCD are displayed in Figure 4.

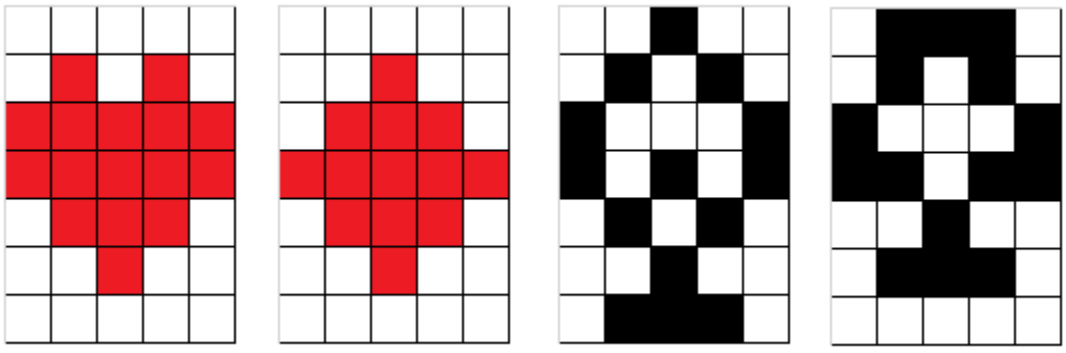

Figure 3: Custom Symbol Mock-Up

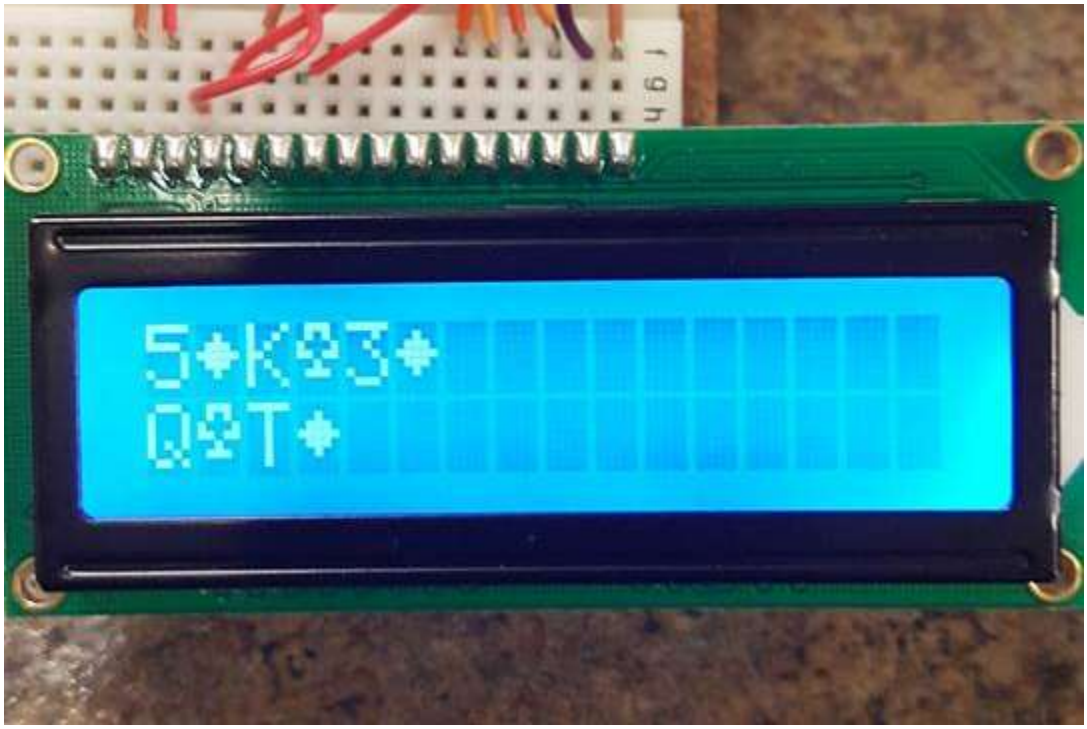

Figure 4: Symbols Displayed on LCD Screen

3.3 Setting Bet with the Potentiometer

At the start of each round the user starts at a screen on the LCD that shows their balance and current bet. Then using the potentiometer they are able to change the bet anywhere from 1 to your balance (all-in). Then the user presses a button to start the game. The code for the betting is at the start of the loop. The potentiometer can be in position 0 to 1023. The team declared a variable potVal and the potentiometer read its current position. Then the team mapped all of the positions from 0 to balance. Then the LCD displays the bet and continues doing this until the button to start the game is pressed.

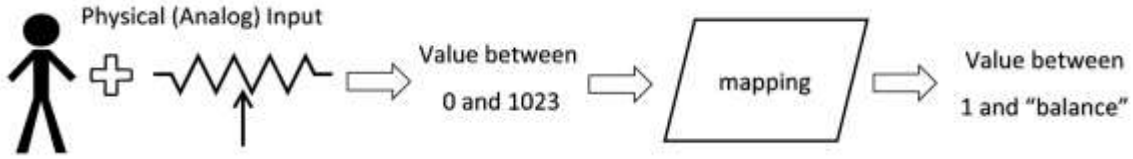

Figure 5: Input/Output Diagram for Potentiometer Use

3.4 User Input

We made two versions of our code. One that works 100% of the time and uses user input straight from the laptop and another that uses buttons. For the user input we originally planned to have two buttons. One button would be used for the player to select “hit” and the other would be for the player to “stand”. The team coded it to read the button status and hit or stand when appropriate. The team was able to get it to hit every time the player pressed hit; however, some of the time when you pressed stand it would also hit. This appears to be some bug in the system because after hours of debugging we are unable to find an error in our code or with the hardware setup. For the computer input code we used a Serial.read and then compared it to the letters ‘h’ and ‘s’.

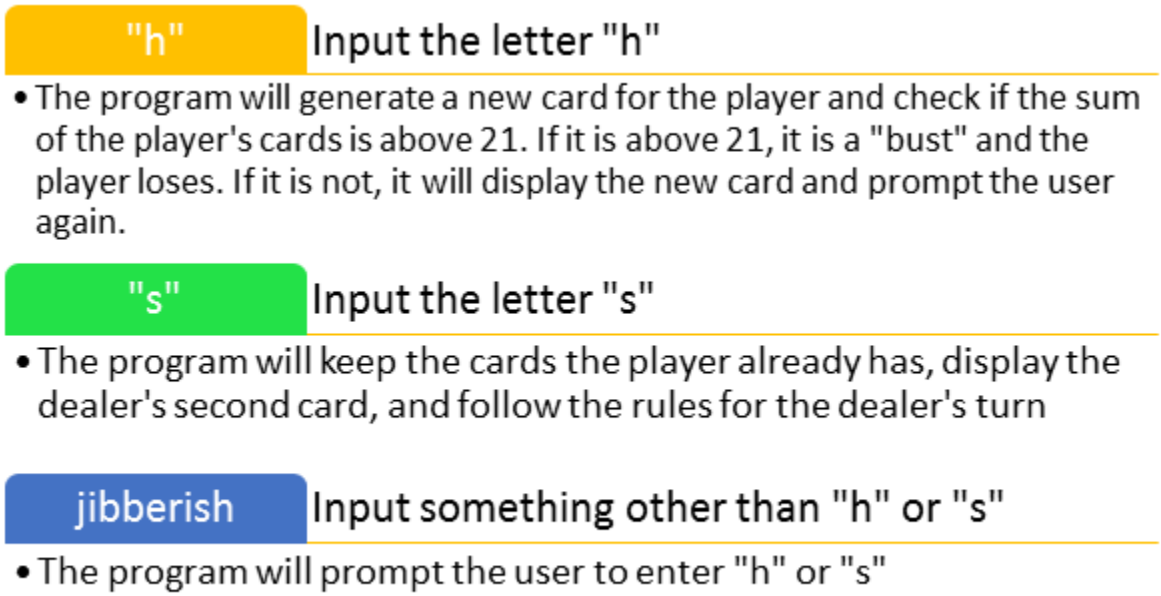

Figure 6: Input/Output Diagram for User Input

3.5 Dealing

As a collective group we worked on the deal function. When called the deal function will be given a true or false statement to determine which player it is dealing to and also a card number which tells it where to put the card in the player/dealer's hand. Before each game we re-initialize an array called deck. This deck is (A-2-3-4-5-6-7-8-9-T-J-Q-K) 4 times as in a standard playing deck. When the deal function is called a random number between 0-51 is generated and that "card" is taken from the deck and its position in the array is made NULL. Then it checks which suit the card is by comparing it to four different ranges of 13 (0-12,13-25….etc). Last, the suit and card character are returned so that they can be printed to the LCD screen. A picture of this can be found above in Figure 4.

4. Results and Screenshots/Photos

With the exception of the buttons that the group ended up abandoning in favor of user input directly to the computer (as pictured in Figure 8 below), the Arduino Blackjack project went very well. Originally, the group broke down the project into the following tasks:

Task	**Task Due Date**	**Task Allocation**
Prepare project proposal	December 2nd	All (using google docs)
Wire the Arduino with the LCD, buttons, and potentiometer	December 4th	Samantha and Braeden
Code the potentiometer, testing out mapping values in order to use it to change betting amounts- must be able to change depending on how much money the player has left	December 4th	Samantha
Code the basic parameters of blackjack (ie. deck array, how the deck is dealt each round, rules that the dealer follows)	December 8th	Samantha

Code conditionals for scoring (how to score aces, exit round if player's score is above 21)	December 8th	Braeden and Brooks
Code the user inputs in blackjack (ie. buttons)	December 8th	Samantha and Brooks
Code for the LCD display	December 10th	Brooks
Finish coding and debugging the blackjack game	December 10th	All- TBD based on what is left
Connect to arduino and start running tests	December 11th	All
Prepare presentation	December 12th	All (using google docs and meeting)

Figure 7: Task Allocation Chart

By December 6th, we were done with all the assignments through December 8th as well as coding for the LCD display. Unfortunately, we realized our program did not run properly due to an error with the buttons. First, the group checked and checked again to make sure that the buttons were wired correctly. The group also used a digital multimeter to ensure that the buttons were not in some way broken or shorted. Furthermore, the group attempted to switch out the buttons for other buttons and even attempted to wire them to a different breadboard. All of these attempts were to no avail, so the group settled on having the player interface with the game through the computer keyboard. Still, until the last possible moment, the group worked together to comb through every line of code seeking a mistake to explain why the buttons were not functioning as intended and potentially to fix them. As illustrated below, our final product did not feature any buttons; however, it still featured a potentiometer to give an analog dimension to the interaction of the project with players. As evidenced by the photos in Figure 9, after completing the project, the group ran the program over and over again in order to ensure that every possible scenario that we attempted to account for through the code was in fact accounted for.

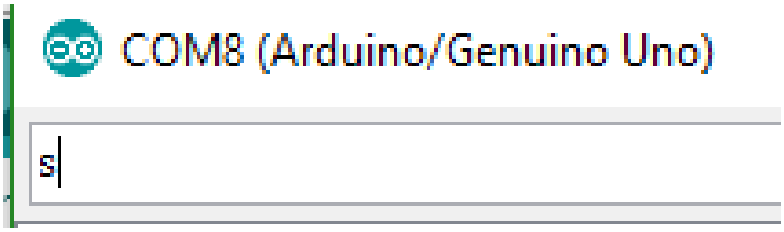

Figure 8: “Stay” Input

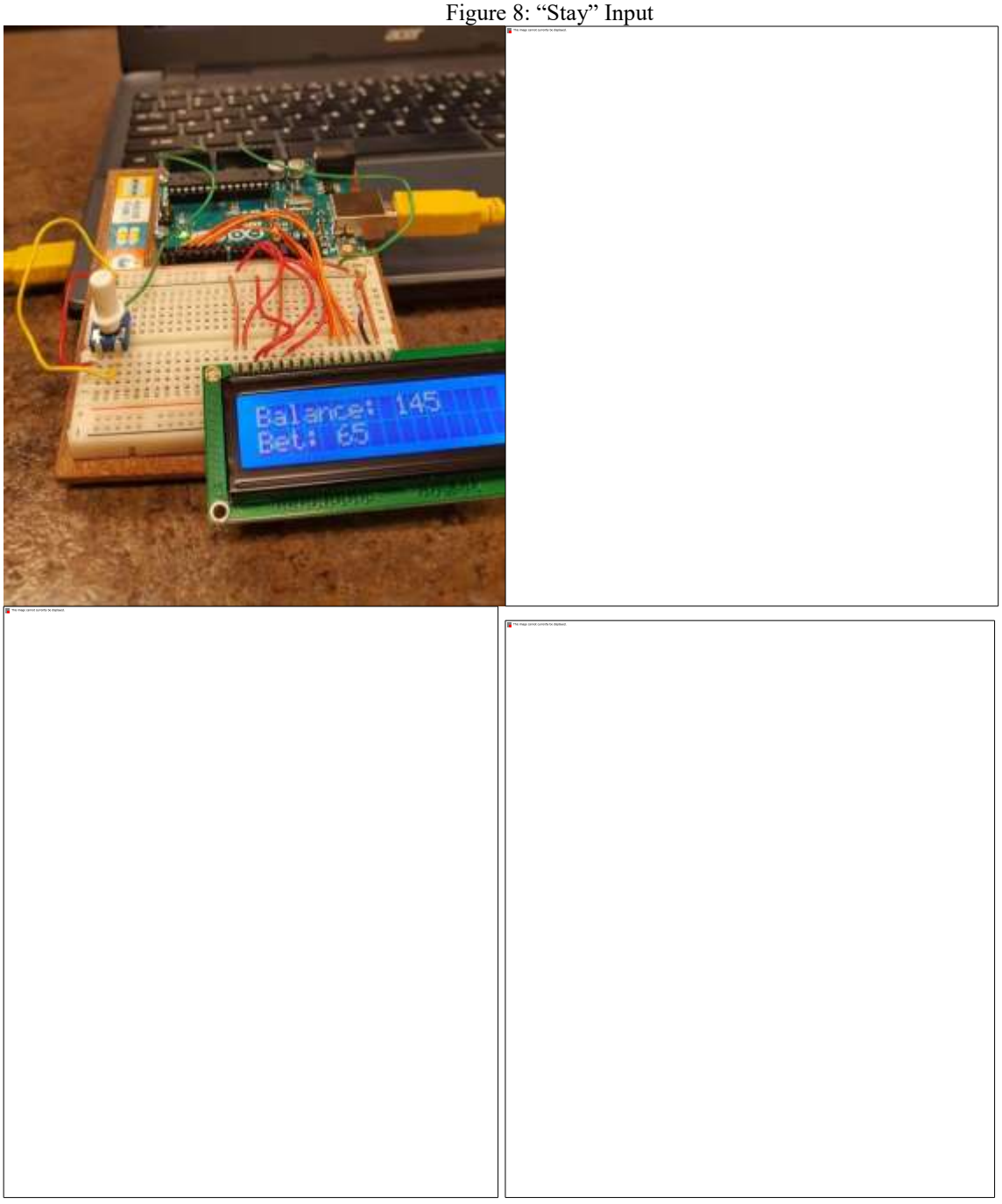

Figure 9: Various Views of Gameplay on the Arduino

5. Discussion and Future Work

The group achieved its goal of creating a fully functional blackjack interactive game, although it was not set up the way it was originally intended. With the absence of the buttons, the user cannot interact only with the arduino board itself but must also interface with the computer's keyboard. Although this was not what was originally intended, the group found how beneficial it is to plan ahead for a contingency plan. Because the group did so, they were able to produce a working game with very little setback. In the future, students could try to actually make this program work with buttons used in order to "hit" or "stay." Students could also add more players to the game- whether computer generated or user-interactive. Alternatively, students could use more than one deck or "shuffle" an array once per gameplay and go through it in order so players could potentially count cards.