Milestone 8- Final Project

✨Flashing Lights ✨

Huzaifa Javed -> hjaved7 -> hjaved7@uic.edu
Aditya Nadkarni -> anadka4 -> anadka4@uic.edu
Raghav Sharma -> rsharm48 -> rsharm48@uic.edu
Nilesh Jain -> njain36 -> njain36@uic.edu

Abstract:

The project is a memorization game in which there is a 2x3 LED light pattern that is to be memorized by the player and then outputted in the same sequence from the controller board. Another board will have all of the necessary information such as a high score, difficulty, user info, displayed on a screen while the final board outputs sound according to the game events occurring. The project will communicate between all 4 boards to allow a user to progress between progressively larger sequences until the user fails.

Overall Description:

The project involves using the Arduino to display a sequential pattern of LED's flashing as a memory game. It will use a form of Rx and Tx to communicate with another arduino board. The user then plays the game by trying to input the same pattern as the flashing led lights in the correct order to move on further in the game. For this idea we would need input devices such as switches to turn on the game when it is ready to take input from the phone as well as a button to adjust difficulty. We will use led lights as output devices to display the pattern, an LCD screen to show users the difficulty and a buzzer which will interact with the user depending on the inputs. What makes this project unique is the utility of wireless communication for a matching pattern game that would normally exist all on one single device. The purpose of the system is to create a fun and addicting game for all ages. The game uses a masterboard(1) with a 2x3 LED light sending information to the display(3) and buzzer(4) boards. These 2 boards are more for output purposes while the input comes from controller(2) board where the user provides input back to the board.

Description of each subsystem.

Arduino & BreadBoard 1:
The first board is the master board with the light sequence being displayed. In this board, a 2x3 array of LED lights will be placed and each one will be paired with a number value. The values of the board will be stored and shown on board 3. A counter to increase the difficulty will also be set in place to increase the number of lights that flash in the sequence entailing a longer memorization.

Arduino & BreadBoard 2:
The second board is the controller board where you enter the memorized bits using buttons and you send this input to the master board using it's connection to the arduino which we can wirelessly send information from, to the master board. The highest score will be sent to the third board.

Arduino & BreadBoard 3:
The third board is the LCD board where the games start button exists and there is an LCD that has relevant information such as difficulty, scores, and current user input being sent. This board's arduino receives information from board 1 and 2 and uses that information to display the correct level and score on the LCD.

Arduino & BreadBoard 4:
The fourth board is a buzzer that commentates on the current game. It will give feedback to the player and announce if they were correct or incorrect. It will also have special announcements about whether they have won or lost and other game related information.

Device (Board 1)	**Behavior**	**Output**
Start Button	Causes the device to start up and initiates the random value generator.	LED Lights

Device (Board 2)	**Behavior**	**Output**

Joystick	Used to read if the user to input up, down, left, right, neutral, or the corners	LED Light

Device (Board 2)	**Behavior**	**Output**
Button	Used to select the user desired LED	LED Light

Device (Board 3)	**Behavior**	**Output**
Joystick	Used as a cursor on the LCD for multiple purposes such as entering name	LCD

Device (Board 3)	**Behavior**	**Output**
Button	This is for entering the input selected by the joystick	LCD

Device (Board 4)	**Behavior**	**Output**
Joystick Input	This is supposed to interact with the user according to the input entered	buzzer

Communication Mechanism:

The controller board, LCD screen board, and buzzer board will be connected and communicating to the master/game board via Rx and Tx.

Controller → Master/Game Board

The controller board will be connected to the master board via Bluet and will send the input instructions (button and joystick) to the master board. It will also have a start/end button that will send messages to the master board to start or end the game.

Master/Game Board → LCD Screen

The master board will be connected to the LCD screen board which will show the user the information about this game like the level of difficulty, score, etc.

Master/Game Board → buzzer

The buzzer will constantly announce to the user their decisions and game state which will be received from the master board.

Master/Game Board → LED Board

The master board will send the different puzzle sequence information to the LED board after which they will read the data and blink accordingly. They will change their speed depending on the difficulty level. Once the user observes the sequence that was shown, he/she gets to choose which LED position should be selected and they will blink accordingly. Once selected, the information will be sent to the master board and it will follow the next steps.

Similar Projects :

https://create.arduino.cc/projecthub/Jerepondumie/make-an-arduino-memory-game-73f55e

Statement of Originality vs 1st Project:

This project is different from our own because it only uses one arduino. This game is also a lot more simple and has a primitive layout of LEDs while our project has a 2x3 layout for more complex game logic and more difficulty making the game more addicting since people are inclined to want to solve a more difficult task.

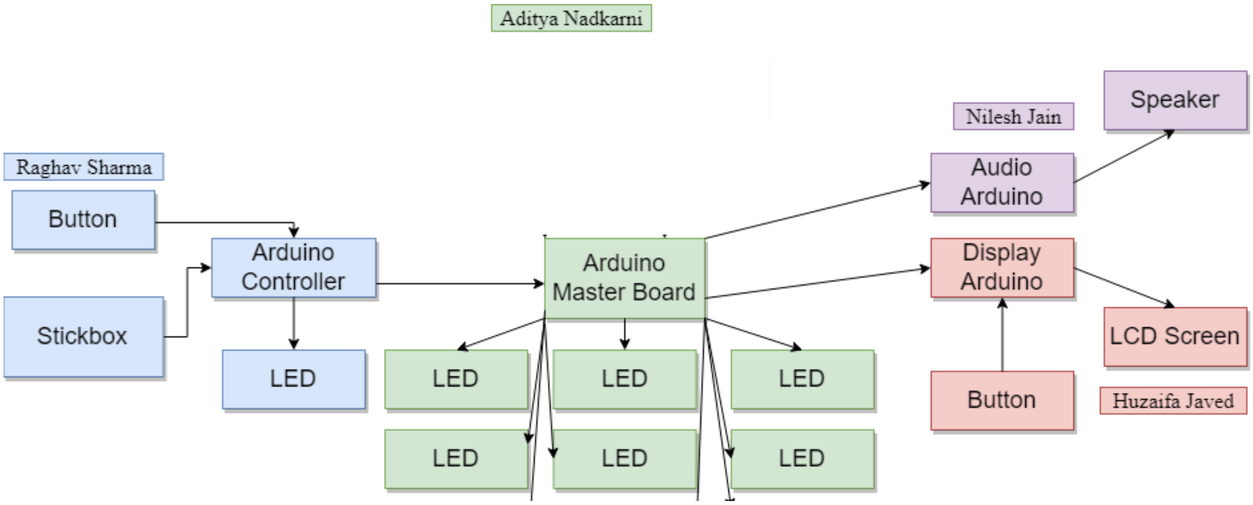
https://maker.pro/arduino/projects/how-to-build-an-arduino-based-memory-game

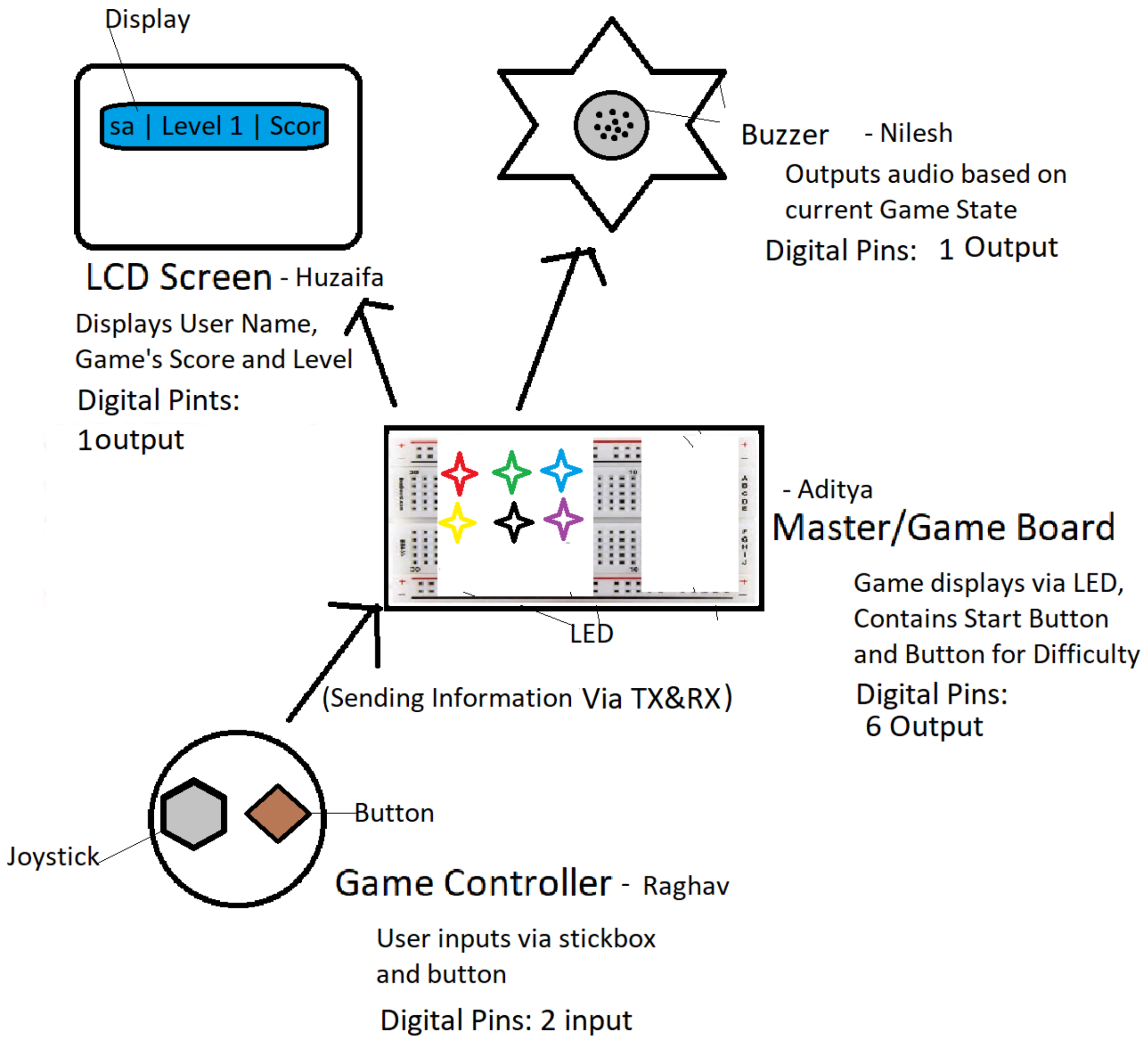
Statement of Original Work vs 2nd Project:

The reason this project is unique is due to multiple reasons. Our project is unique because it uses an analog stick as opposed to buttons. The buzzer and the lcd board have multiple functionalities that are also something that haven’t been implemented in this project as well.

Material List:

~ 50 wires
~ 25 resistors
10 LED lights
5 Buttons
3 Joysticks
2 LCD displays
2 potentiometers
1 buzzer
4 Rx and Tx transmission Wires





LCD Screen - Huzaifa

Displays User Name, Game's Score and Level

Digital Pints: 1output

Buzzer - Nilesh

Outputs audio based on current Game State

Digital Pins: 1 Output

- Aditya

Master/Game Board

Game displays via LED, Contains Start Button and Button for Difficulty

Digital Pins: 6 Output

(Sending Information Via TX&RX)

Game Controller - Raghav

User inputs via stickbox and button

Digital Pins: 2 input

GND and Power/5V Pins would be used for all Arduinos

Problems and Solutions:

- The breadboard did not fit the LEDs in a 3x3 array so we reduced it to a 2x3, This made us re-adjust our controller to cycle through the LEDs linearly rather than implementing commands with circular motions.
- Could not achieve a speaker so a buzzer was used as a replacement thus we used different tones to represent the feedback rather than vocal feedback.
- Connected boards together 1 to 1 with Rx and Tx as opposed to Bluetooth connection
  - Made the wires very messy and unfeasible for transportation
  - Easy disconnections

1. Steps to build your project

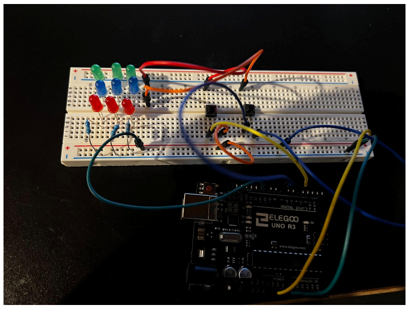
Discussion on how to build your project (think Lab Reports)
The Arduino will be assembled in order of MasterBoard-> Controller -> Display -> Audio.
For the master board we start on this primarily because we need the 2x3 LED lights to be functional standalone first. The Masterboard will also need to include wiring for the Rx and Tx of each other board to connect to it. Once the code is implemented using a random seed to generate a number in an array to light up a respective LED, After the board can generate inputs on command, the second board will be made to provide input and connect with Rx and Tx. Using an analog stick, the second board will use a test LED to move around which LED will light up. Once the test LED has served its purpose the communication method of Rx and Tx will be used to allow the same input to apply to the master board. After the testing of this main functionality is complete, both the 3rd display board and the 4th board can be made simultaneously. Since both of these boards are just outputs of information and event-driven, we can use code to show what to display based on the current progress of the person using the device. The audios are built in reaction to the user getting a level correct or incorrect.

2. User guide

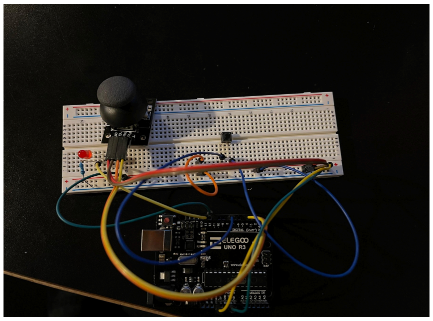
The user has a very simple and seamless process. The user would walk into the system already built and connected to a computer. The user would first use the LCD board and use the joystick to enter their name by going downward to increase to the next letter in the alphabet. When reaching the desired letter, they click the button on the LCD board to input the selected letter. After this all they have to do is click the start button on the main arduino. They use the controller board's joysticks and the button to select the pattern on the main board after it flashes. The buzzer board will work asynchronously without intervention from the user and the user can press the button on this buzzer board to listen to the instructions. After this process, the user can enjoy the game.

Video Presentation: https://youtu.be/VvDcWGEY1WY

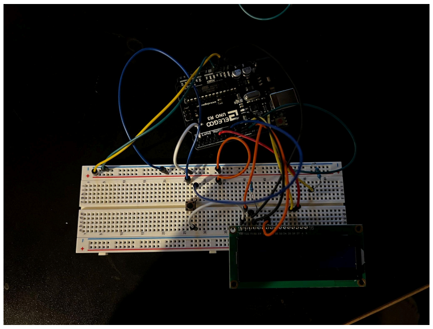
Arduino & BreadBoard 1:



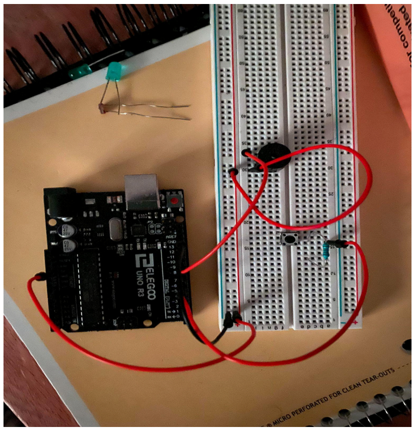
Arduino & BreadBoard 2:



Arduino & BreadBoard 3:

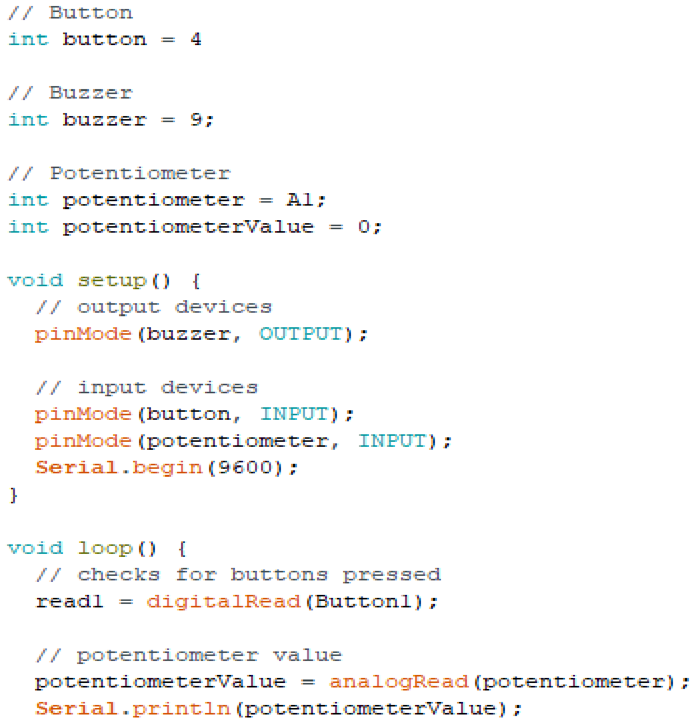


Arduino & BreadBoard 4:

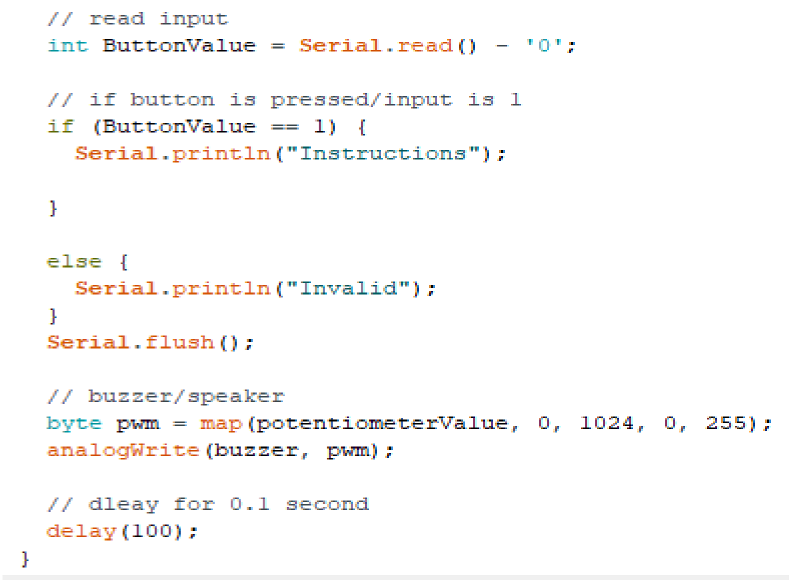


Code:

We have independent chunks of code for some of our boards but have not been able to put everything together at the moment. This is the most complete code at the moment.

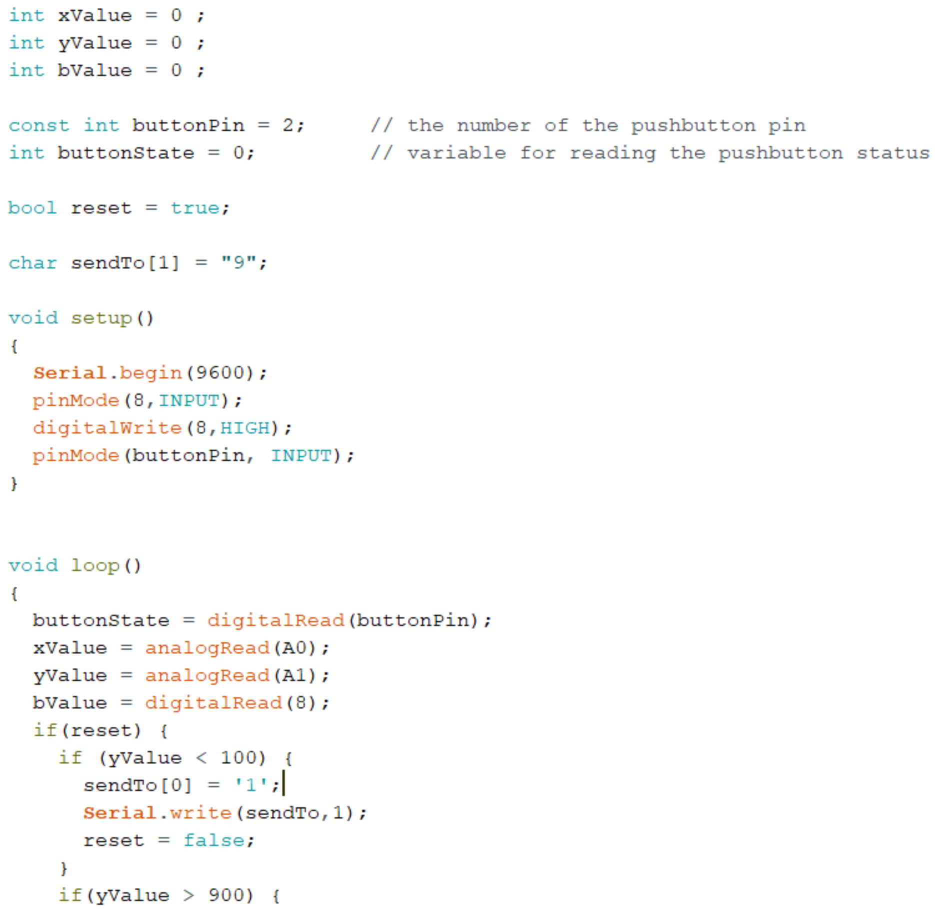


```
// Button
int button = 4

// Buzzer
int buzzer = 9;

// Potentiometer
int potentiometer = A1;
int potentiometerValue = 0;

void setup() {
  // output devices
  pinMode(buzzer, OUTPUT);

  // input devices
  pinMode(button, INPUT);
  pinMode(potentiometer, INPUT);
  Serial.begin(9600);
}

void loop() {
  // checks for buttons pressed
  read1 = digitalRead(Button1);

  // potentiometer value
  potentiometerValue = analogRead(potentiometer);
  Serial.println(potentiometerValue);
```



```
  // read input
  int ButtonValue = Serial.read() - '0';

  // if button is pressed/input is 1
  if (ButtonValue == 1) {
    Serial.println("Instructions");

  }

  else {
    Serial.println("Invalid");
  }
  Serial.flush();

  // buzzer/speaker
  byte pwm = map(potentiometerValue, 0, 1024, 0, 255);
  analogWrite(buzzer, pwm);

  // dleay for 0.1 second
  delay(100);
}
```

Controller Code:



```
int xValue = 0 ;
int yValue = 0 ;
int bValue = 0 ;

const int buttonPin = 2;     // the number of the pushbutton pin
int buttonState = 0;         // variable for reading the pushbutton status

bool reset = true;

char sendTo[1] = "9";

void setup()
{
  Serial.begin(9600);
  pinMode(8,INPUT);
  digitalWrite(8,HIGH);
  pinMode(buttonPin, INPUT);
}


void loop()
{
  buttonState = digitalRead(buttonPin);
  xValue = analogRead(A0);
  yValue = analogRead(A1);
  bValue = digitalRead(8);
  if(reset) {
    if (yValue < 100) {
      sendTo[0] = '1';
      Serial.write(sendTo,1);
      reset = false;
    }
    if(yValue > 900) {
      sendTo[0] = '0';
      Serial.write(sendTo,1);
      reset = false;
    }
    if (buttonState == HIGH) {
      sendTo[0] = '2';
      Serial.write(sendTo,1);
      reset = false;
    }
    if (bValue == LOW) {
      sendTo[0] = '3';
      Serial.write(sendTo,1);
      reset = false;
    }
  }
  if(!reset) {
    if(yValue > 400 && yValue < 600 && buttonState == LOW && bValue == HIGH) {
      reset = true;
    }
  }

  delay(1);
}
```

Master Board Code:

```
int count = 0;
int count2 = 0;
char input[2];
int mode = 3;
int index = 2;
int myArrayAns[100];
int myArray[100];
int counter = 0;
char output[2];


void setup() {
  Serial.begin(9600);
}

// the loop function runs over and over again forever
void loop() {

  int countArr = 0;
   Serial.readBytes(input,1);
   Serial.println(input);
    if(mode == 3) {
      if(input[0] == '3') {
```

```
   output[0] = '8';
   Serial.write(output, 1);
   mode = 0;
   count = 1;
   input[0] = '9';
  }
 } else {
  if(input[0] == '3') {
  digitalWrite(index, LOW);
  delay(500);

  for(int x = 0; x < count; x++) {
   if (myArrayAns[x] != myArray[x]) {
       mode = 4;
       output[0] = '7';
       Serial.write(output, 1);
       output[0] = '9';
       break;

   }
  }
  mode = 0;
  input[0] = '9';
  if (count == 6) {
   output[0] = '6';
   Serial.write(output, 1);
   output[0] = '9';
   count = 0;
  }
 }
}


if(mode == 1) {
 digitalWrite(index, HIGH);
 if(input[0] == '0'){
  digitalWrite(index, LOW);
  index++;
  if(index == 8) {
   index = 2;
  }
  digitalWrite(index, HIGH);
  input[0] = '9';
 }
 if(input[0] == '1') {
  digitalWrite(index, LOW);
  index--;
  if(index == 1) {
   index = 7;
```

```
      }
      digitalWrite(index, HIGH);
      input[0] = '9';
    }
    if(input[0] == '2') {
      myArrayAns[count2] = index;
      input[0] = '9';
      count2++;
    }
  }

  if (mode == 0) {
  for (int i = 0; i < count; i++) {
     int randNumber = random(2,7);
     myArray[i] = randNumber;
     digitalWrite(randNumber, HIGH);   // turn the LED on (HIGH is the voltage level)
     delay(500);                       // wait for a second
     digitalWrite(randNumber, LOW);    // turn the LED off by making the voltage LOW
     delay(500);
  }
  count2 = 0;
  count++;
  mode = 1;
  }

  delay(1);                       // wait for a second


}
```

Week:	**Planning:**
Eight (Done)	Do research on proposed aspects of the project in milestone 3. Find tutorials, and gather information on how the project will function and any errors we might need to account for. Post all research and project tweaks into the updated design document.
Nine (Done)	Gather all the missing materials and keep doing research. Each person has been assigned a board and they will conduct research on each aspect of their respective board.

Ten (Done)	Start building Master Board and developing logic for the puzzle such as randomizing sequences and pairing values to the LED Lights
Eleven (Done)	Milestone 4 due; Start building controller board. The research from the earlier weeks should be sufficient to understand how to send the information from this board to the master board. If this is not the case, we will split up the work and do the physical and research aspects in pairs.
Twelve	Milestone 5 due; Start building the display board and confirm all the data between each. At this point we should be well versed in communication between boards so the physical and communication aspects should not be challenging. Do extensive testing between the two boards and see if they work without errors.
Thirteen	Start building a Sound Board and implementing the new piece whether that be a buzzer.. (Optional: add custom voice lines for certain game events)
Fourteen	Milestone 5 II due; Polish up code as well as make sure the program works as a full fledged game through extensive testing.
Fifteen	All Milestones are completed. The Video of the final fully functional project is completed.
Sixteen	Final due and already completed.

References

“Blinking Led with Arduino Uno.” *Arduino Project Hub*, https://create.arduino.cc/projecthub/RoyB/blinking-led-with-arduino-uno-01e098.

“Make an Arduino Memory Game.” *Arduino Project Hub*, https://create.arduino.cc/projecthub/Jerepondumie/make-an-arduino-memory-game-73f55e.

Wijesinghe, Jinani. “How to Build an Arduino-Based Memory Game: Arduino.” *Maker Pro*, Maker Pro, 26 Feb. 2022, https://maker.pro/arduino/projects/how-to-build-an-arduino-based-memory-game.