

TinkerCad-Link: <https://www.tinkercad.com/things/g5fogUMh1Oi-magnificent-bojo/editel>

### Task:

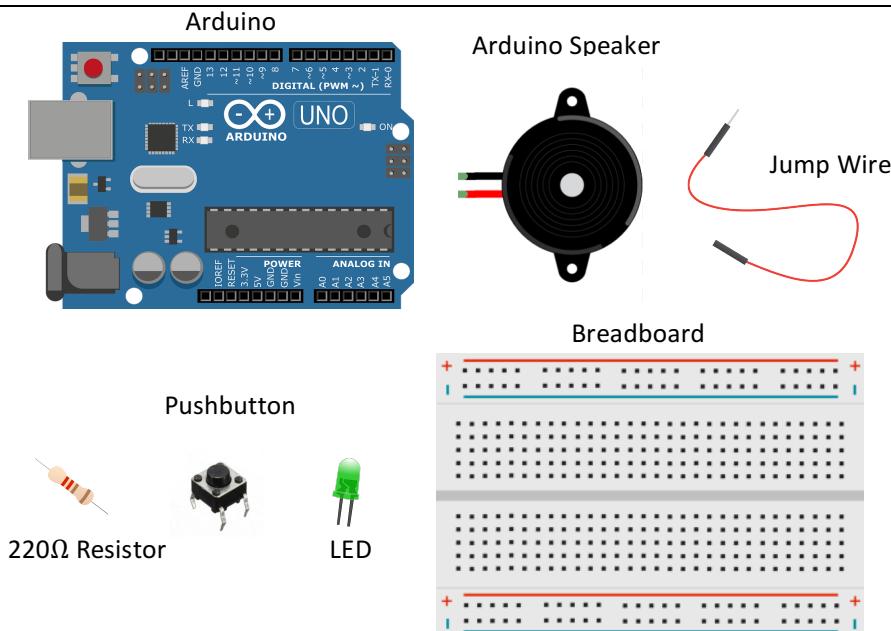
---

To design and build an Arduino implementation of the classic Simon Says memory game. In this game, a random colored pattern flashes on four LEDs, and it is the job of the player to enter that pattern on four corresponding switches. The pattern starts at length 1 (i.e. only a single LED ash), and increases in length by one each time the user enters the pattern correctly. The game ends (and resets) when the user makes a mistake. The pattern should be randomly generated.

### Materials:

---

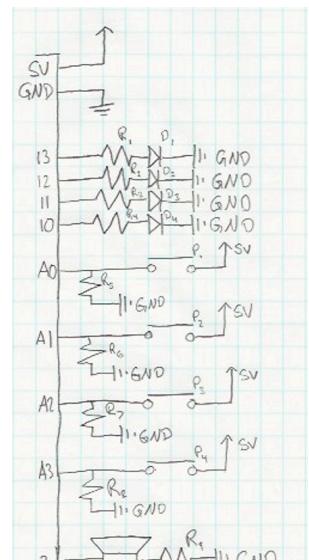
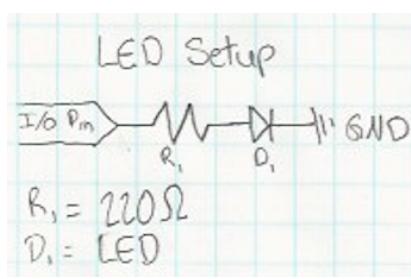
| Item:           | Quantity |
|-----------------|----------|
| Arduino         | 1        |
| 220Ω Resistor   | 9        |
| Pushbutton      | 4        |
| Red LED         | 1        |
| Yellow LED      | 1        |
| Blue LED        | 1        |
| Green LED       | 1        |
| Breadboard      | 1        |
| Arduino Speaker | 1        |
| Jump Wires      | 17       |



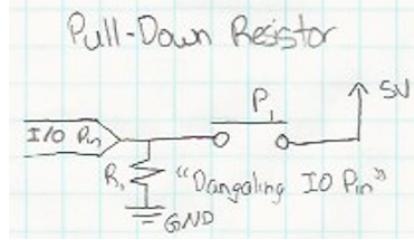
### Procedure:

---

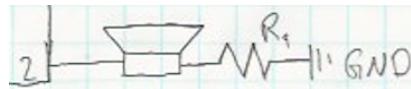
1. Create Schematic for breadboard and all components.
  - a. Schematic for Project detailed to right →
  - b. To Include
    - i. Resistor (R1-4) -> LED (D1-4) -> GND (4 times)
      1. 220Ω Resistor was used due to the LED using 2V.



ii. Pull down Resistors for pushbuttons (4 times)



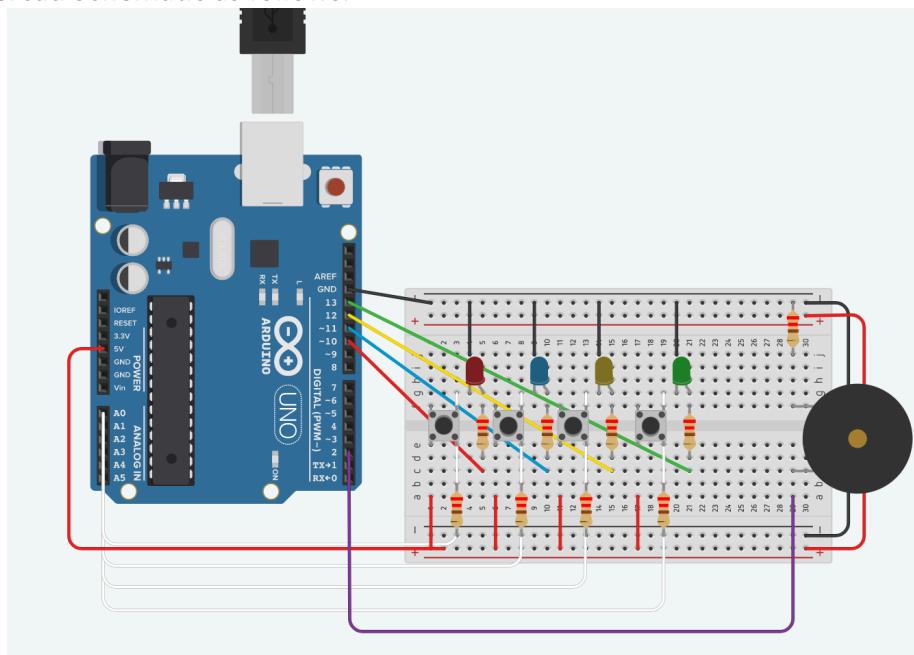
iii. 220Ω Resistor -> Arduino-Speaker -> GND



1. A 220Ω Resistor was used to lower the volume of the speaker to allow for beeps and light flash awareness to be at an appropriate level.

2. Using Tinkercad/circuits execute and create the previous schematic.

a. Tinkercad Schematic as follows.



3. Progressively develop code to execute Simon Says.

a. Further Detailed in Software Design Below.

4. Test and add components accordingly.

### **Software Design:**

Software for this assignment is broken into 3 general states, after an initialization that will ask the user for input to generalize a user play level.

The software design follows as such:

#### **Initialization:**

- Declare global variables; such as the sequence array/difficulty/level and state.
- Setup Pins A0-A3 to be input pins for pushbuttons
- Setup Pins 10-13 to be output pins for pushbuttons
- Create a sequence of LED flashes waiting for user to hold red button to begin

Iteratively:

```
get_difficulty
    - Wait for user input for level of difficulty, depending on which LED is chosen from the breadboard.
show_sequence:
    - Show sequence until current user level
get_sequence:
    - Get user input for the previous sequence shown
    - if Correct Sequence
        - flash correct sequence pattern
if Incorrect Pattern:
    - Flash incorrect pattern message
    - Break loop and start back again from beginning of Iteratively
```

These methods and functions were chosen due to the simplicity and functionality it would create for the project.

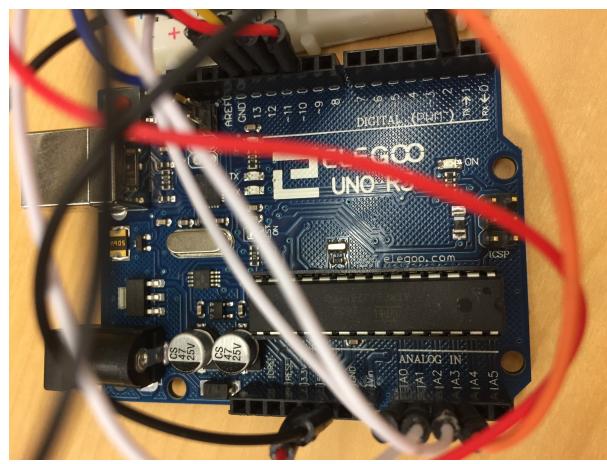
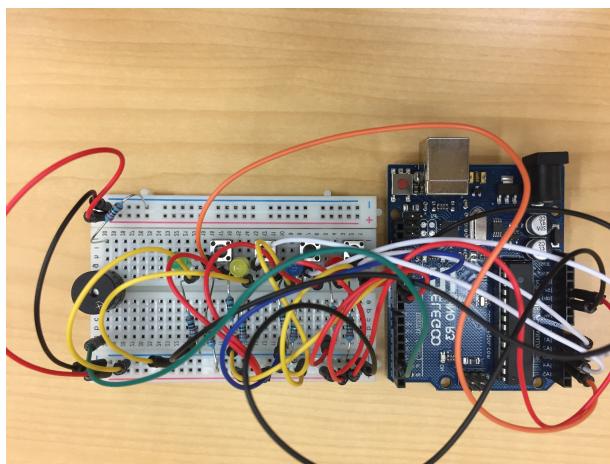
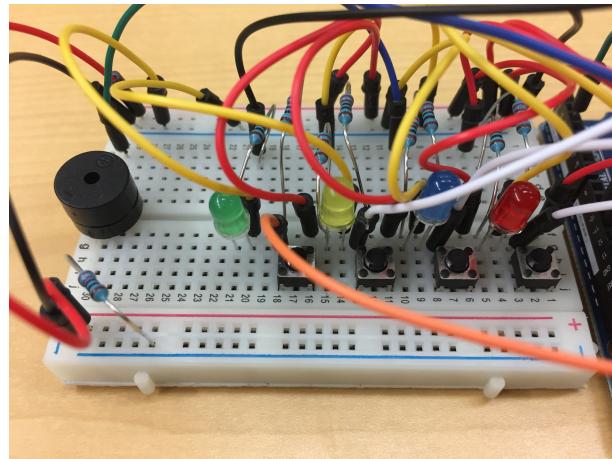
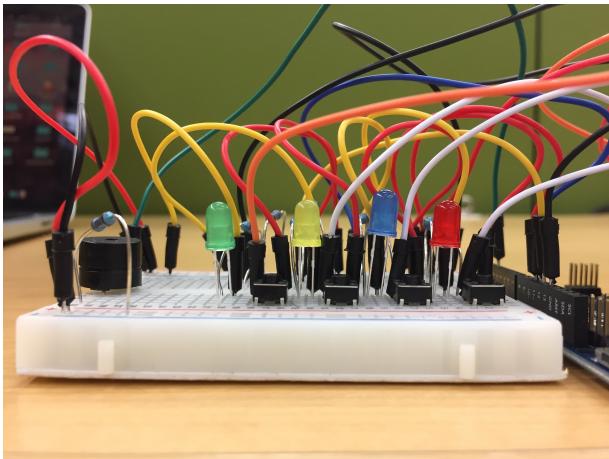
#### **IMPLEMENTATION/EXECUTION:**

---

1. I chose to begin the game with a backwards and forwards pattern of LED flashes with their accompanying sound to give the user a chance to see which LED sounded like, this will increase their ability to remember and repeat patterns.
2. Once the begin button is held the LED's will flash 3 times at the fourth flash they will all stay illuminated until the user chooses their desired difficulty.
3. Once a difficulty is chosen that light will immediately turn off and pause to give the user a chance to remove their finger from the board and prepare for the sequence.
4. Then the sequence will be shown, and if the user correctly inputs the sequence provided the program will quickly flash all LED's 3 times along with a high pitch noise to notify the user they correctly input the sequence provided.
5. Step 4 will be continuously be repeated with one more LED added into the pattern until the user makes a mistake.
6. If the user makes a mistake the program will blink all 4 LED's 3 times and play a low sound to notify the user that their sequence was incorrect.
7. It will then illuminate all 4 LED's and wait for the user to select a new level of play before showing the next sequence of patterns.

## Results/Conclusion:

---



The final project for this assignment resulted with a working Arduino/breadboard combination, that will successfully play the game Simon says.

## ADDED-FEATURES:

---

1. Lighted sequencing waiting for the user to hold the start button.
2. Added 4 different levels of difficulty that the user has the ability to choose before they play each game.
3. Speaker that will follow all the lights as they are illuminated on the breadboard, including when the user inputs information on the breadboard.
4. Lights will illuminate when the user presses the corresponding pushbutton.
5. Correct and Incorrect LED/Speaker patterns when the user either completes a sequence or incorrectly enters sequence.

## DISCONTINUITYS:

---

1. TinkerCad schematic and actual hardware created for this project do not perfectly align. Pushbuttons on TinkerCad are limited to span across 4 pins, whereas on the breadboard created for this project to increase usability the pushbutton pins were altered to span 3 pins.