THE MEMORY GAME

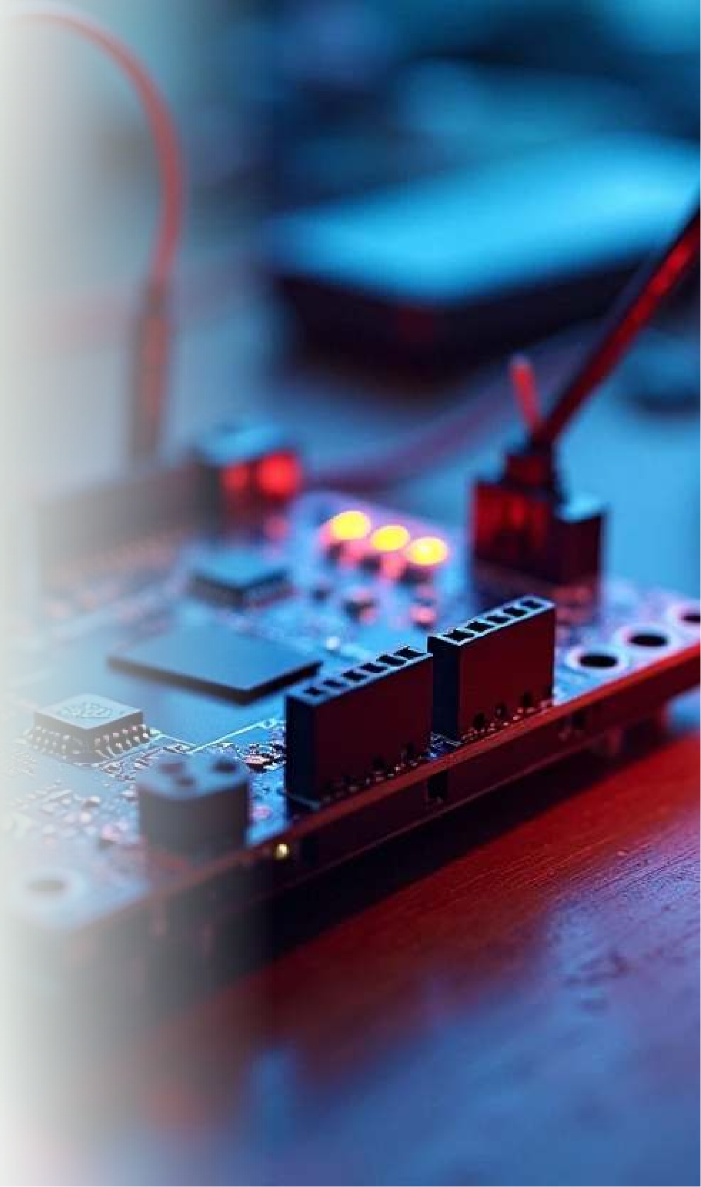
By-Mohammad Faris

Introduction

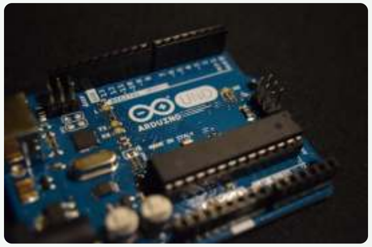
This project is a simple memory-based Simon Says game built using an Arduino Uno, LEDs, buttons, an LCD, and a buzzer. The game challenges the player to repeat a sequence of lights and sounds that becomes longer with each level. It includes two game modes — Progressive and Random — and a sound toggle option. This project demonstrates basic concepts of Arduino programming, including input handling, output control, and randomization.

This project demonstrates key concepts in embedded systems, including:

- Digital input/output
- Analog input reading
- Random number generation
- User feedback via sound and display
- Control structures and game logic



Components Required



Arduino Uno

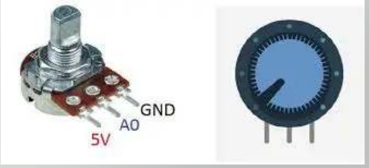
Main microcontroller board to control all components.

LEDs (x4)

Light-emitting diodes used for blinking patterns.

Push Button(x4)

User input to switch between LED patterns.



Potentiometer

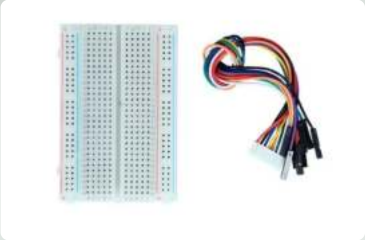
Variable resistor to control LED blinking speed.

16×2 LCD

Detects ambient light level.

Buzzer

Produces sound during intruder alert.



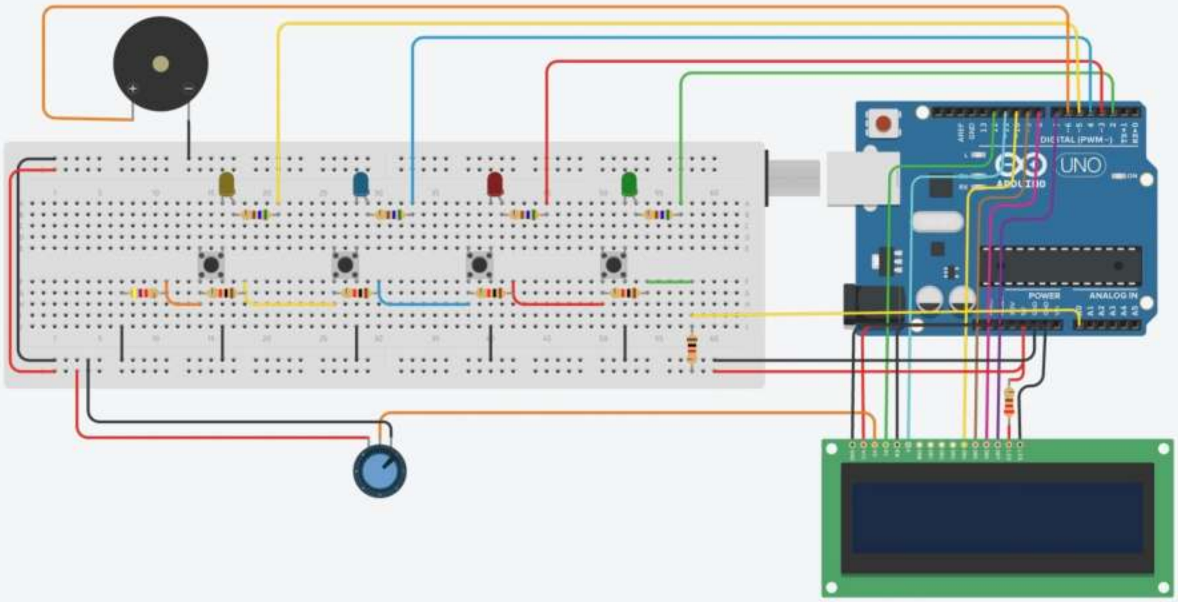
Breadboard & Jumper Wires

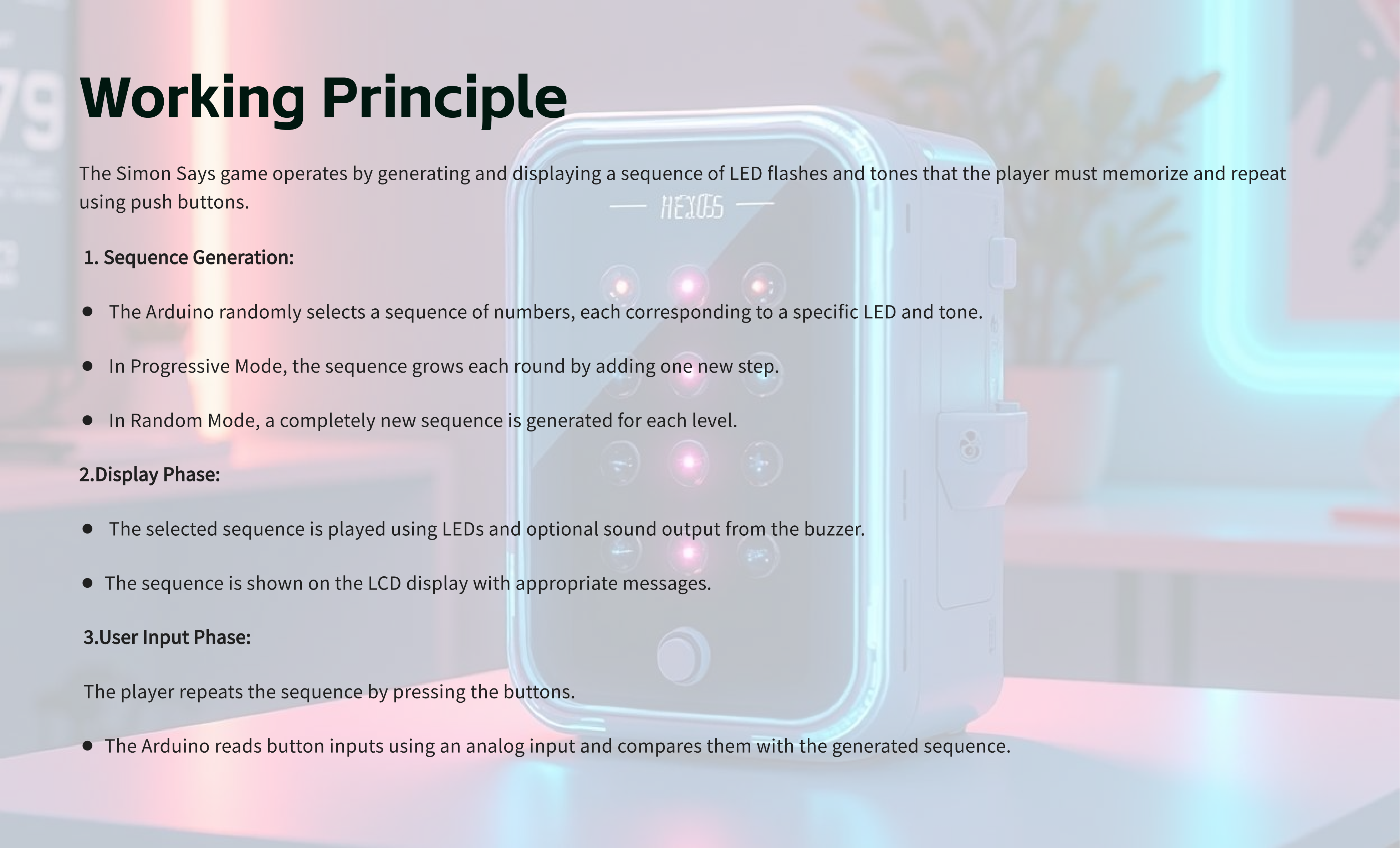
For connecting components without soldering.

Resistors

Limit current for LEDs and sensors

Circuit Diagram





Working Principle

The Simon Says game operates by generating and displaying a sequence of LED flashes and tones that the player must memorize and repeat using push buttons.

1. Sequence Generation:

- The Arduino randomly selects a sequence of numbers, each corresponding to a specific LED and tone.
- In Progressive Mode, the sequence grows each round by adding one new step.
- In Random Mode, a completely new sequence is generated for each level.

2.Display Phase:

- The selected sequence is played using LEDs and optional sound output from the buzzer.
- The sequence is shown on the LCD display with appropriate messages.

3.User Input Phase:

The player repeats the sequence by pressing the buttons.

- The Arduino reads button inputs using an analog input and compares them with the generated sequence.

4.Validation:

- If the input matches the sequence, the player moves to the next level.
- If not, the game ends and shows the final score.

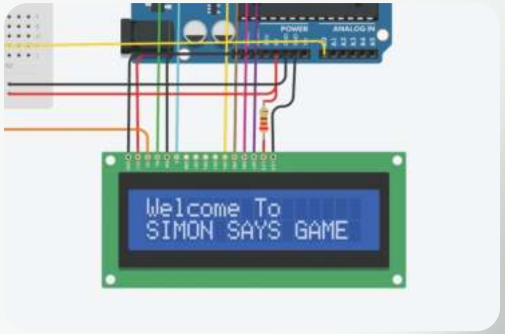
4.Feedback:

- Correct input: brief success tone or LED flash.
- Wrong input: game over message, buzzer alert, and score display

Code Snippet

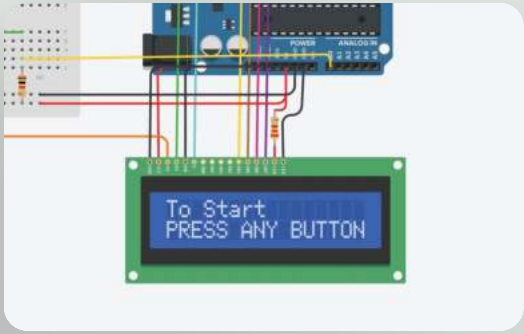
```
int seq[50], level = 1;

void loop() {

 // Generate sequence

  for (int i = 0; i < level; i++)

    seq[i] = random(1, 5);

  // Show sequence to player

  for (int i = 0; i < level; i++) {

    lightLed(seq[i]);

    playTone(seq[i]);

    delay(300);

    lightLed(0); // Turn off LEDs

    delay(200);
  }
  // Get user input and check
  for (int i = 0; i < level; i++) {
    int input = getButton();
    if (input != seq[i]) {
      gameOver();
      level = 1;
      return;
    }
  }
  level++; // Advance to next level
  delay(1000);
}
```

LCD Display (Text Output)



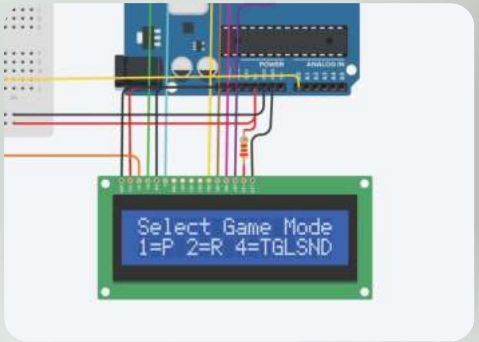
```
Welcome To
SIMON SAYS GAME
```

Welcome messages ("SIMON SAYS GAME").



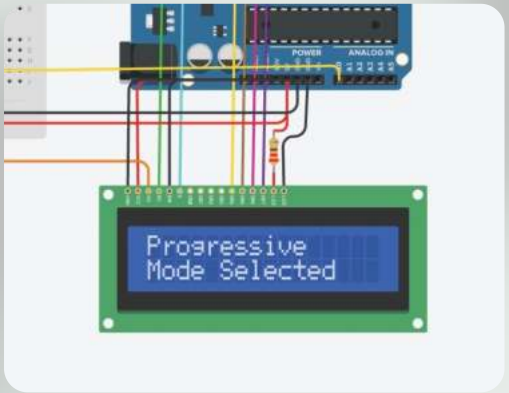
```
To Start
PRESS ANY BUTTON
```

Instructions ("Press Any Button", "Your Turn", etc.).



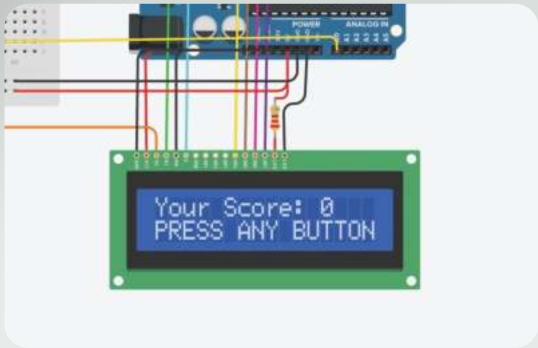
```
Select Game Mode
1=P 2=R 4=TGLSND
```

Game mode selection prompt.



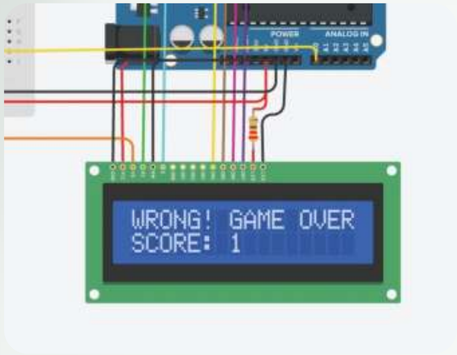
```
Progressive
Mode Selected
```

Current level display.



```
Your Score: 0
PRESS ANY BUTTON
```

Game over message.

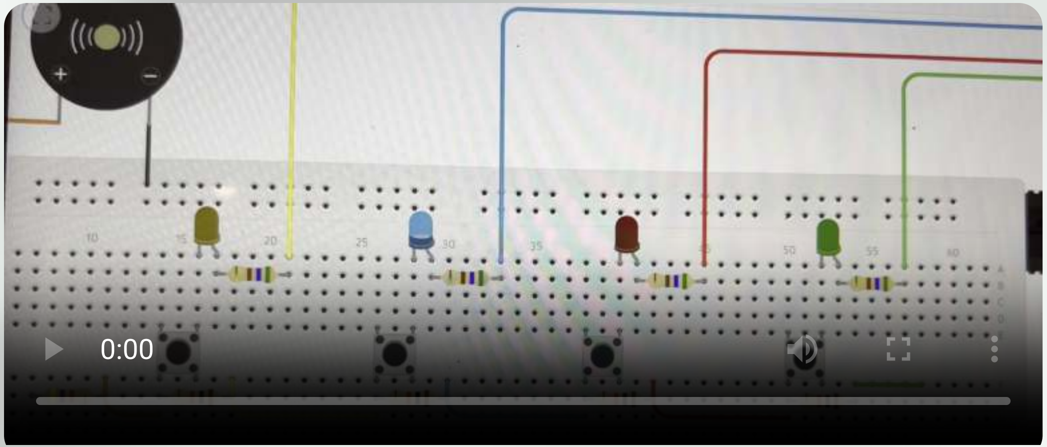


```
WRONG! GAME OVER
SCORE: 1
```

Final score

- **LEDs (Visual Output)**

Flashing LEDs to show the sequence.



- # 🎵 Buzzer Functions

Event	Buzzer Output
Game Start	A single tone (confirmation)
LED Step Played	A specific tone per LED (1–4)
Correct Input	Brief tone for each correct press
Level Complete	3 quick success tones
Game Over	5 sharp error tones
Sound Toggle ON/OFF	Confirmation tone when toggled

Applications

S.No	Application Area	Description
1	🧠 Memory Training	Improves short-term memory and concentration through pattern repetition.
2	👦 Educational Tool	Teaches logic, sequencing, and basic electronics to students.
4	🔧 Embedded Systems Learning	Helps beginners understand Arduino programming and hardware interfacing.
5	🎉 Entertainment	● Used as a fun game in events, homes, or hobby projects.

- # Project Conclusion

- The project successfully recreates the classic Simon Says memory game using Arduino.
- It combines hardware (LEDs, buzzer, LCD) and software logic for interactive gameplay.
- Enhances understanding of input/output operations, timing, and program flow.
- Offers a fun way to improve memory skills and user interaction handling.
- Useful for educational purposes, hobby projects, and learning embedded systems.

THANK YOU