

Mind Game

1. Project Title:

Mind Game Using Arduino UNO

2. Abstract:

The Mind Game is an Arduino-based interactive electronic game designed to test a player's memory, focus, and response speed.

The game works on multiple levels where LEDs glow in a random sequence along with unique sounds. The player must remember and repeat the same sequence using push buttons.

If the player presses the correct sequence, the game moves to the next level and shows success using an RGB LED, buzzer, and LCD display. If the player fails, the game shows a failure indication and restarts. After completing all levels, a celebration mode with lights and sound is activated.

3. Objectives of the Project:

- To design an interactive memory-based game
- To understand Arduino programming and logic building
- To learn hardware–software integration
- To use LCD display for real-time feedback
- To implement random sequence generation
- To improve understanding of digital electronics and embedded systems

4. Components Required:

<u>Components</u>	<u>Quantity</u>
• Arduino UNO	1
• LEDs	5
• Push Button Switches	5
• Common Anode RGB LED	1
• Buzzer	1
• 16X2 I2C LCD Display	1
• Resistors(220 ohm)	8
• Jumper Wires	As needed

5. Block Diagram Description:

The Arduino UNO acts as the main controller.

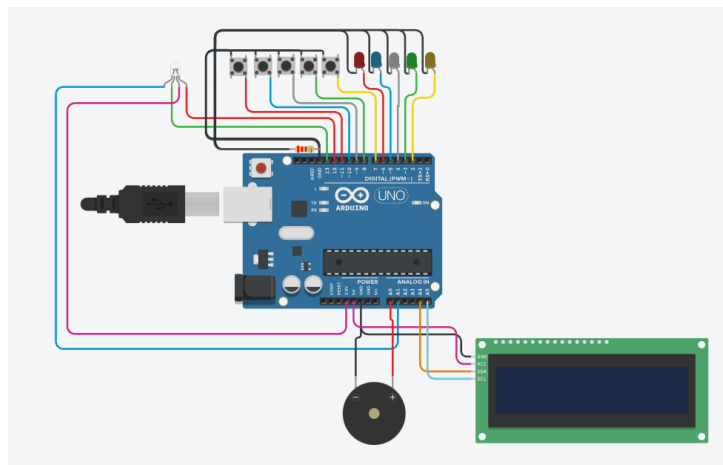
- LEDs and buzzer provide visual and audio output
- Push buttons take player input
- LCD displays level number and game status
- RGB LED shows success, failure, and celebration modes

6. Pin Configuration:

- LEDs:

LED1	D2
LED2	D3
LED3	D4
LED4	D5
LED5	D6
- Push Buttons:

- | | |
|-----------------------------------|-----|
| Button 1 | D7 |
| Button 2 | D8 |
| Button 3 | D9 |
| Button 4 | D10 |
| Button 5 | D11 |
| • <u>RGB LED:</u> | |
| Red | D12 |
| Green | D13 |
| Blue | A1 |
| Common Anode | 5V |
| • <u>Other Components:</u> | |
| Buzzer | A0 |
| LCD SDA | A4 |
| LCD SCL | A5 |
| LCD VCC | 5V |
| LCD GND | GND |



7. Arduino Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define NUM_LEDS 5
#define NUM_LEVELS 10

int ledPins[NUM_LEDS] = {2, 3, 4, 5, 6};
int buttonPins[NUM_LEDS] = {7, 8, 9, 10, 11};

// RGB LED (COMMON ANODE)
int redPin = 12; // RED = FAIL
int greenPin = 13; // GREEN = SUCCESS
int bluePin = A1; // BLUE = CELEBRATION

int buzzer = A0;

// Unique tones for each LED
int ledTones[NUM_LEDS] = {400, 500, 600, 700, 800};


int sequence[NUM_LEVELS];
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
void setup() {  
  Serial.begin(9600);  
  
  lcd.init();  
  lcd.backlight();  
  lcd.clear();  
  lcd.setCursor(0, 0);  
  lcd.print("WELCOME TO");  
  delay(1500);  
  lcd.clear();  
  lcd.print("Mind Reader Game");  
  delay(2000);  
  lcd.clear();  
  lcd.print("ARE YOU READY?");  
  delay(2500);  
  lcd.clear();  
  
  for (int i = 0; i < NUM_LEDS; i++) {  
    pinMode(ledPins[i], OUTPUT);  
    pinMode(buttonPins[i], INPUT_PULLUP);  
  }
```


```
  pinMode(redPin, OUTPUT);  
  pinMode(greenPin, OUTPUT);  
  pinMode(bluePin, OUTPUT);  
  pinMode(buzzer, OUTPUT);
```

```
  rgbOff(); // Turn RGB OFF  
  randomSeed(analogRead(0));  
}
```

```
void loop() {  
  for (int level = 1; level <= NUM_LEVELS; level++) {  
  
    generateSequence(); //  New random sequence every level
```

```
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Level: ");  
    lcd.print(level);  
    delay(800);
```

```
    showSequence(level);
```

```
    if (!checkPlayer(level)) {  
      //  FAILED  
      lcd.clear();  
      lcd.setCursor(0, 0);  
      lcd.print("Level ");
```

```

    lcd.print(level);
    lcd.setCursor(0, 1);
    lcd.print("FAILED!");

    rgbRed();    // 🟡 RGB RED
    lostSound();
    delay(1200);
    rgbOff();

    level = 0; // Restart from Level 1
    continue;
}

// ✓ SUCCESS
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Level ");
lcd.print(level);
lcd.setCursor(0, 1);
lcd.print("SUCCESS!");

    rgbGreen();    // 🟢 RGB GREEN
    successSound();
    delay(1200);
    rgbOff();
}

// 🎉 PLAYER COMPLETED ALL LEVELS
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("All Levels Clear");
lcd.setCursor(0, 1);
lcd.print("WINNER!");

celebrationMode();
while (1);
}

// -----
void generateSequence() {
    for (int i = 0; i < NUM_LEVELS; i++) {
        sequence[i] = random(NUM_LEDS);
    }
}

// -----
void showSequence(int level) {
    for (int i = 0; i < level; i++) {
        int led = sequence[i];

        digitalWrite(ledPins[led], HIGH);

```

```

    tone(buzzer, ledTones[led], 300);
    delay(350);

    digitalWrite(ledPins[led], LOW);
    delay(250);
}
}

// -----
bool checkPlayer(int level) {
    for (int i = 0; i < level; i++) {
        int pressed = waitForButtonPress();
        if (pressed != sequence[i]) {
            return false;
        }
    }
    return true;
}

// -----
int waitForButtonPress() {
    while (true) {
        for (int i = 0; i < NUM_LEDS; i++) {
            if (digitalRead(buttonPins[i]) == LOW) {

                digitalWrite(ledPins[i], HIGH);
                tone(buzzer, ledTones[i], 250);
                delay(250);
                digitalWrite(ledPins[i], LOW);

                return i;
            }
        }
    }
}

// -----
void successSound() {
    tone(buzzer, 1000, 200);
    delay(150);
    tone(buzzer, 1500, 200);
}

void lostSound() {
    tone(buzzer, 300, 300);
    delay(300);
    tone(buzzer, 200, 300);
}

// -----
void celebrationMode() {

```

```

for (int x = 0; x < 30; x++) {

    rgbRed();
    tone(buzzer, 800, 100);
    delay(120);

    rgbGreen();
    tone(buzzer, 1000, 100);
    delay(120);

    rgbBlue();
    tone(buzzer, 1200, 100);
    delay(120);

    rgbOff();
}
}

// -----
// ☒ COMMON ANODE RGB CONTROL (LOW = ON, HIGH = OFF)

void rgbRed() {
    digitalWrite(redPin, LOW);
    digitalWrite(greenPin, HIGH);
    digitalWrite(bluePin, HIGH);
}

void rgbGreen() {
    digitalWrite(redPin, HIGH);
    digitalWrite(greenPin, LOW);
    digitalWrite(bluePin, HIGH);
}

void rgbBlue() {
    digitalWrite(redPin, HIGH);
    digitalWrite(greenPin, HIGH);
    digitalWrite(bluePin, LOW);
}

void rgbOff() {
    digitalWrite(redPin, HIGH);
    digitalWrite(greenPin, HIGH);
    digitalWrite(bluePin, HIGH);
}

```

8. **Working Principle:**

- When the game starts, a welcome message is shown on the LCD.
- The game has 10 levels.
- In each level:
 - ✓ A new random LED sequence is generated

- ✓ LEDs glow with unique sound tones
- The player presses buttons to repeat the sequence:
 - ✓ Correct input → Green color on RGB LED + success sound
 - ✓ Wrong input → Red color on RGB LED + failure sound
- If the player completes all 10 levels:
 - ✓ Celebration mode activates
 - ✓ RGB LED flashes like disco lights
 - ✓ Buzzer plays celebration sound
 - ✓ LCD shows “*Winner*”

9. **Features of the Project:**

- Multi-level game (10 levels)
- New random sequence in every level
- Unique sound for each LED
- LCD-based level and status display
- RGB LED for success, failure, and celebration
- Interactive and user-friendly design

10. **Applications:**

- Educational learning tool
- Memory and brain training game
- School and college project
- Robotics and STEM demonstrations
- Embedded systems practice project

11. **Advantages:**

- Easy to understand and use
- Low cost and easily available components
- Improves logical thinking
- Interactive audio-visual feedback
- Expandable with more levels or displays

12. **Limitations:**

- Limited number of LEDs
- No score storage
- Single-player game
- No wireless control

13. **Future Enhancements:**

- Add score system
- Add difficulty modes (Easy/Medium/Hard)
- Add OLED or TFT display
- Add Bluetooth or mobile app control
- Multiplayer mode

- Timer-based challenge

14. **Conclusion:**

The Mind Game using Arduino is a fun and educational project that combines electronics, programming, and creativity. It successfully demonstrates the use of Arduino for interactive game development while improving understanding of memory-based logic and embedded systems. This project is suitable for students, beginners, and hobbyists interested in electronics and robotics.