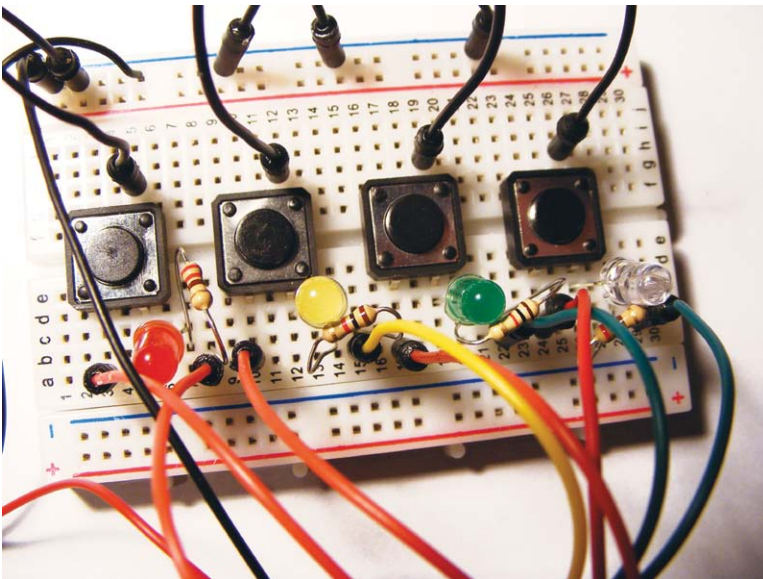
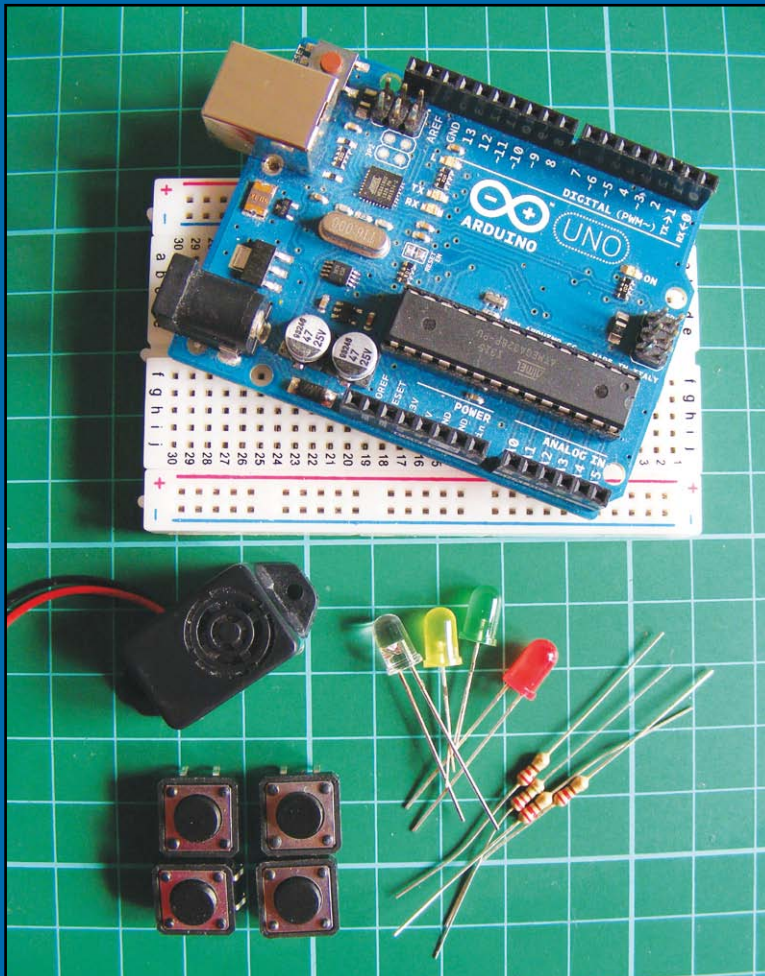


PROJECT 8: MEMORY GAME

IN THIS PROJECT WE'LL CREATE OUR OWN VERSION OF AN ATARI ARCADE MEMORY GAME CALLED TOUCH ME, USING FOUR LEDS, FOUR PUSHBUTTON SWITCHES, A PIEZO BUZZER, AND SOME RESISTORS AND JUMPER WIRES.





PARTS REQUIRED

- Arduino board
- Breadboard
- Jumper wires
- Piezo buzzer
- 4 momentary tactile four-pin pushbuttons
- 4 LEDs
- 4 220-ohm resistors

LIBRARIES REQUIRED

- Tone

HOW IT WORKS

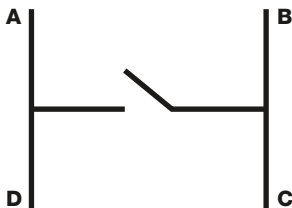
The original Atari game had four colored panels, each with an LED that lit up in a particular pattern that players had to repeat back (see Figure 8-1).



This memory game plays a short introductory tune and flashes an LED. When you press the correct corresponding button, the lights flash again in a longer sequence. Each time you repeat the sequence back correctly, the game adds an extra step to make the sequence more challenging for you. When you make an error, the game resets itself.

THE BUILD

1. Place the pushbuttons in the breadboard so they straddle the center break with pins A and B on one side of the break, and C and D on the other, as shown in Figure 8-2. (See Project 1 for more information on how the pushbutton works.)



2. Connect pin B of each pushbutton to the GND rail of your breadboard, and connect the rail to Arduino GND.
3. Connect pin D of each pushbutton to Arduino's digital pins 2 through 5 in order.

FIGURE 8-1:

The original
Touch Me game

FIGURE 8-2:

A pushbutton has four pins.

- Insert the LEDs into the breadboard with the shorter, negative legs connected to pin C of each pushbutton. Insert the positive leg into the hole on the right, as shown in the circuit diagram in Figure 12-3.

PUSHBUTTON	ARDUINO/LED
Pin B	GND
Pin C	LED negative legs
Pin D	Arduino pins 2–5

- Place a 220-ohm resistor into the breadboard with one wire connected to the positive leg of each LED. Connect the other wire of the resistor to the Arduino as follows.

LEDS	ARDUINO/ PUSHBUTTON
Positive legs	Arduino pins 8–11 via 220-ohm resistors
Negative legs	Pushbutton pin C

Make sure the red LED connected to pin 11 is paired with the pushbutton connected to pin 5, the yellow LED connected to pin 10 is paired with the pushbutton connected to pin 4, the green LED connected to pin 9 is paired with the pushbutton connected to pin 3, and the blue LED connected to pin 8 is paired with the pushbutton connected to pin 2.

- Connect the black wire of the piezo directly to Arduino GND, and the red wire to Arduino pin 12.

PIEZO	ARDUINO
Red wire	Pin 12
Black wire	GND

- Check your setup against Figure 8-3, and then upload the code in “The Sketch” on page 73.

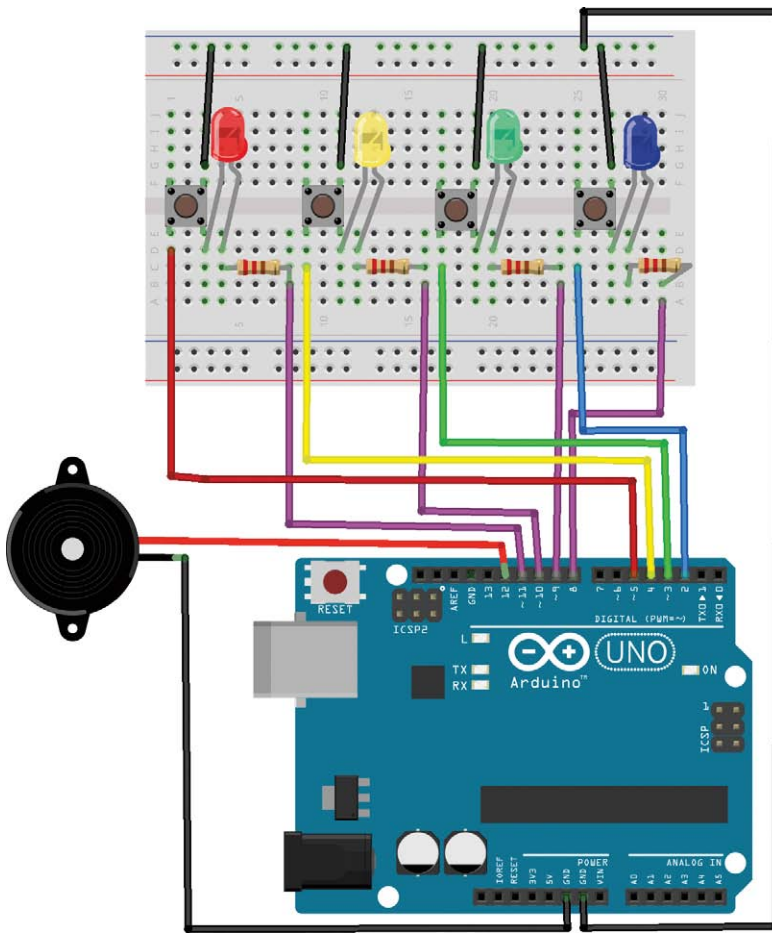


FIGURE 8-3:
Circuit diagram for
the memory game

THE SKETCH

The sketch generates a random sequence in which the LEDs will light; a random value generated for y in the pattern loop determines which LED is lit (e.g., if y is 2, the LED connected to pin 2 will light). You have to follow and repeat back the pattern to advance to the next level.

In each level, the previous lights are repeated and one more randomly generated light is added to the pattern. Each light is associated with a different tone from the piezo, so you get a different tune each time, too. When you get a sequence wrong, the sketch restarts with a different random sequence. For the sketch to compile correctly, you will need to install the Tone library (available from <http://nostarch.com.com/arduinohandbook/>). See “Libraries” on page 7 for details.

```

-----
// Used with kind permission from Abdullah Alhazmy www.Alhazmy13.net

#include <Tone.h>
Tone speakerpin;
int starttune[] = {NOTE_C4, NOTE_F4, NOTE_C4, NOTE_F4, NOTE_C4,
                  NOTE_F4, NOTE_C4, NOTE_F4, NOTE_G4, NOTE_F4,
                  NOTE_E4, NOTE_F4, NOTE_G4};
int duration2[] = {100, 200, 100, 200, 100, 400, 100, 100, 100, 100,
                  200, 100, 500};
int note[] = {NOTE_C4, NOTE_C4, NOTE_G4, NOTE_C5, NOTE_G4, NOTE_C5};
int duration[] = {100, 100, 100, 300, 100, 300};
boolean button[] = {2, 3, 4, 5}; // Pins connected to
                                // pushbutton inputs
boolean ledpin[] = {8, 9, 10, 11}; // Pins connected to LEDs
int turn = 0; // Turn counter
int buttonstate = 0; // Check pushbutton state
int randomArray[100]; // Array that can store up to 100 inputs
int inputArray[100];

void setup() {
  Serial.begin(9600);
  speakerpin.begin(12); // Pin connected to piezo buzzer
  for (int x = 0; x < 4; x++) {
    pinMode(ledpin[x], OUTPUT); // Set LED pins as output
  }
  for (int x = 0; x < 4; x++) {
    pinMode(button[x], INPUT); // Set pushbutton pins as inputs
    digitalWrite(button[x], HIGH); // Enable internal pullup;
                                // pushbuttons start in high
                                // position; logic reversed
  }
  // Generate "more randomness" with randomArray for the output
  // function so pattern is different each time
  randomSeed(analogRead(0));
  for (int thisNote = 0; thisNote < 13; thisNote++) {
    speakerpin.play(starttune[thisNote]); // Play the next note
    if (thisNote == 0 || thisNote == 2 || thisNote == 4 ||
        thisNote == 6) { // Hold the note
      digitalWrite(ledpin[0], HIGH);
    }
    if (thisNote == 1 || thisNote == 3 || thisNote == 5 ||
        thisNote == 7 || thisNote == 9 || thisNote == 11) {
      digitalWrite(ledpin[1], HIGH);
    }
    if (thisNote == 8 || thisNote == 12) {
      digitalWrite(ledpin[2], HIGH);
    }
    if (thisNote == 10) {
      digitalWrite(ledpin[3], HIGH);
    }
    delay(duration2[thisNote]);
    speakerpin.stop(); // Stop for the next note
    digitalWrite(ledpin[0], LOW);
  }
}

```

```

    digitalWrite(ledpin[1], LOW);
    digitalWrite(ledpin[2], LOW);
    digitalWrite(ledpin[3], LOW);
    delay(25);
}
delay(1000);
}

void loop() {
    // Generate the array to be matched by the player
    for (int y = 0; y <= 99; y++) {
        digitalWrite(ledpin[0], HIGH);
        digitalWrite(ledpin[1], HIGH);
        digitalWrite(ledpin[2], HIGH);
        digitalWrite(ledpin[3], HIGH);
        // Play the next note
        for (int thisNote = 0; thisNote < 6; thisNote++) {
            speakerpin.play(note[thisNote]); // Hold the note
            delay(duration[thisNote]);      // Stop for the next note
            speakerpin.stop();
            delay(25);
        }
        digitalWrite(ledpin[0], LOW);
        digitalWrite(ledpin[1], LOW);
        digitalWrite(ledpin[2], LOW);
        digitalWrite(ledpin[3], LOW);
        delay(1000);
        // Limited by the turn variable
        for (int y = turn; y <= turn; y++) {
            Serial.println("");
            Serial.print("Turn: ");
            Serial.print(y);
            Serial.println("");
            randomArray[y] = random(1, 5); // Assign a random number (1-4)
            // Light LEDs in random order
            for (int x = 0; x <= turn; x++) {
                Serial.print(randomArray[x]);
                for (int y = 0; y < 4; y++) {
                    if (randomArray[x] == 1 && ledpin[y] == 8) {
                        digitalWrite(ledpin[y], HIGH);
                        speakerpin.play(NOTE_G3, 100);
                        delay(400);
                        digitalWrite(ledpin[y], LOW);
                        delay(100);
                    }
                    if (randomArray[x] == 2 && ledpin[y] == 9) {
                        digitalWrite(ledpin[y], HIGH);
                        speakerpin.play(NOTE_A3, 100);
                        delay(400);
                        digitalWrite(ledpin[y], LOW);
                        delay(100);
                    }
                    if (randomArray[x] == 3 && ledpin[y] == 10) {
                        digitalWrite(ledpin[y], HIGH);
                    }
                }
            }
        }
    }
}

```

```

        speakerpin.play(NOTE_B3, 100);
        delay(400);
        digitalWrite(ledpin[y], LOW);
        delay(100);
    }
    if (randomArray[x] == 4 && ledpin[y] == 11) {
        digitalWrite(ledpin[y], HIGH);
        speakerpin.play(NOTE_C4, 100);
        delay(400);
        digitalWrite(ledpin[y], LOW);
        delay(100);
    }
}
}
}
input();
}
}

// Check whether input matches the pattern
void input() {
    for (int x = 0; x <= turn; x++) {
        for (int y = 0; y < 4; y++) {
            buttonstate = digitalRead(button[y]); // Check for button push
            if (buttonstate == LOW && button[y] == 2) {
                digitalWrite(ledpin[0], HIGH);
                speakerpin.play(NOTE_G3, 100);
                delay(200);
                digitalWrite(ledpin[0], LOW);
                inputArray[x] = 1;
                delay(250);
                Serial.print(" ");
                Serial.print(1);
                // Check if value of user input matches the generated array
                if (inputArray[x] != randomArray[x]) {
                    fail(); // If not, fail function is called
                }
                x++;
            }
            if (buttonstate == LOW && button[y] == 3) {
                digitalWrite(ledpin[1], HIGH);
                speakerpin.play(NOTE_A3, 100);
                delay(200);
                digitalWrite(ledpin[1], LOW);
                inputArray[x] = 2;
                delay(250);
                Serial.print(" ");
                Serial.print(2);
                if (inputArray[x] != randomArray[x]) {
                    fail();
                }
                x++;
            }
            if (buttonstate == LOW && button[y] == 4) {

```



```

    digitalWrite(ledpin[2], HIGH);
    speakerpin.play(NOTE_B3, 100);
    delay(200);
    digitalWrite(ledpin[2], LOW);
    inputArray[x] = 3;
    delay(250);
    Serial.print(" ");
    Serial.print(3);
    if (inputArray[x] != randomArray[x]) {
        fail();
    }
    x++;
}
if (buttonstate == LOW && button[y] == 5) {
    digitalWrite(ledpin[3], HIGH);
    speakerpin.play(NOTE_C4, 100);
    delay(200);
    digitalWrite(ledpin[3], LOW);
    inputArray[x] = 4;
    delay(250);
    Serial.print(" ");
    Serial.print(4);
    if (inputArray[x] != randomArray[x]) {
        fail();
    }
    x++;
}
}
}
delay(500);
turn++; // Increment turn count
}

// Function used if player fails to match the sequence
void fail() {
    for (int y = 0; y <= 2; y++) { // Flash lights to indicate failure
        digitalWrite(ledpin[0], HIGH);
        digitalWrite(ledpin[1], HIGH);
        digitalWrite(ledpin[2], HIGH);
        digitalWrite(ledpin[3], HIGH);
        speakerpin.play(NOTE_G3, 300);
        delay(200);
        digitalWrite(ledpin[0], LOW);
        digitalWrite(ledpin[1], LOW);
        digitalWrite(ledpin[2], LOW);
        digitalWrite(ledpin[3], LOW);
        speakerpin.play(NOTE_C3, 300);
        delay(200);
    }
    delay(500);
    turn = -1; // Reset turn value to start the game again
}

```
