```
#include <TM1637Display.h>

#define CLK_PIN 2
#define DIO_PIN 3

TM1637Display display(CLK_PIN, DIO_PIN);

const int ledPins[7] = {11, 12, 10, 8, 7, 13, 9};
const int buzzerPin = 4;
const int buttonPin = 5;

int score = 0;
int target = 0;
bool shown = false;

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

  for (int i = 0; i < 7; i++) pinMode(ledPins[i], INPUT);

  pinMode(buttonPin, INPUT);
  pinMode(buzzerPin, OUTPUT);

  display.setBrightness(7);
  display.clear();

  randomSeed(analogRead(A0));

  newTarget();
  showScore();
}

void loop() {
  if (!shown) {
    Serial.print("Target: ");
    Serial.println(targetName());
    shown = true;
  }

  if (buttonPressedStable()) {
    int active = getActiveLED();

    bool correct = (active != -1) && isCorrect(active);
```

```
    if (correct) {
      score++;

      Serial.println("Correct!");
      showScore();

      correctCelebration();
      buzzerwin();
    } else {
      score--;

      buzzerlose();
      Serial.println("Wrong!");

      wrongCelebration();
      showScore();

      delay(250);
    }

    if (score >= 9) {
      showScore();
      bigWinBuzzer();
      bigWinCelebration();
      score = 0;
      showScore();
    }

    if (score <= -9) {
      showScore();
      bigLoseBuzzer();
      bigLoseCelebration();
      score = 0;
      showScore();
    }

    Serial.print("Score: ");
    Serial.println(score);

    showScore();

    newTarget();
  }
}
```

```
void buzzerwin() {
  tone(buzzerPin, 523);
  delay(100);
  noTone(buzzerPin);

  tone(buzzerPin, 659);
  delay(100);
  noTone(buzzerPin);

  tone(buzzerPin, 784);
  delay(100);
  noTone(buzzerPin);

  tone(buzzerPin, 1047);
  delay(300);
  noTone(buzzerPin);
}

void buzzerlose() {
  for (int freq = 2000; freq > 100; freq -= 10) {
    tone(buzzerPin, freq, 10);
    delay(5);
  }

  noTone(buzzerPin);
}

void newTarget() {
  int newColor;

  do {
    newColor = random(0, 3);
  } while (newColor == target);

  target = newColor;
  shown = false;
}

void showScore() {
  int val = abs(score);

  if (val > 9) val = 9;
```

```
  uint8_t segs[4] = {0, 0, 0, 0};

  if (score < 0) segs[0] = 0x40;
  else segs[0] = 0x00;

  segs[1] = display.encodeDigit(val);

  display.setSegments(segs);
}

void correctCelebration() {
  uint8_t allOn[4] = {0x7F, 0x7F, 0x7F, 0x7F};
  uint8_t allOff[4] = {0x00, 0x00, 0x00, 0x00};

  for (int i = 0; i < 4; i++) {
    display.setSegments(allOn);
    delay(120);
    display.setSegments(allOff);
    delay(120);
  }

  showScore();
}

void wrongCelebration() {
  uint8_t noText[4] = {
    0x00,
    0x00,
    0x54,
    0x5C
  };

  display.setSegments(noText);
  delay(500);

  showScore();
}

void bigWinCelebration() {
  uint8_t allOn[4] = {0x7F, 0x7F, 0x7F, 0x7F};
  uint8_t allOff[4] = {0x00, 0x00, 0x00, 0x00};
  uint8_t dashes[4] = {0x40, 0x40, 0x40, 0x40};

  for (int i = 0; i < 8; i++) {
```

```
    display.setSegments(allOn);
    delay(100);
    display.setSegments(allOff);
    delay(100);
  }

  display.setSegments(dashes);
  delay(400);

  showScore();
}

void bigWinBuzzer() {
  int melody[] = {
    392, 523, 659, 784, 1047, 784, 1047, 1319, 1568
  };

  int durations[] = {
    100, 100, 100, 120, 180, 100, 150, 180, 450
  };

  for (int i = 0; i < 9; i++) {
    tone(buzzerPin, melody[i]);
    delay(durations[i]);
    noTone(buzzerPin);
    delay(40);
  }
}

void bigLoseCelebration() {
  uint8_t dashes[4] = {0x40, 0x40, 0x40, 0x40};
  uint8_t allOff[4] = {0x00, 0x00, 0x00, 0x00};

  for (int i = 0; i < 5; i++) {
    display.setSegments(dashes);
    delay(150);
    display.setSegments(allOff);
    delay(150);
  }

  showScore();
}

void bigLoseBuzzer() {
```

```
  int melody[] = {
    988, 740, 587, 440, 330, 247, 196
  };

  int durations[] = {
    180, 180, 180, 220, 250, 300, 500
  };

  for (int i = 0; i < 7; i++) {
    tone(buzzerPin, melody[i]);
    delay(durations[i]);
    noTone(buzzerPin);
    delay(60);
  }

  for (int freq = 500; freq > 80; freq -= 15) {
    tone(buzzerPin, freq);
    delay(12);
  }

  noTone(buzzerPin);
}

String targetName() {
  if (target == 0) return "Red";
  if (target == 1) return "Yellow";
  else return "Green";
}

int getActiveLED() {
  for (int i = 0; i < 7; i++) {
    if (digitalRead(ledPins[i]) == HIGH) return i;
  }

  return -1;
}

bool isCorrect(int i) {
  if (target == 0) return (i == 0 || i == 3 || i == 6);
  if (target == 1) return (i == 1 || i == 4);
  else return (i == 2 || i == 5);
}

bool buttonPressedStable() {
```

```
  if (digitalRead(buttonPin) != LOW) return false;

  delay(30);

  if (digitalRead(buttonPin) != LOW) return false;

  while (digitalRead(buttonPin) == LOW) {
    delay(5);
  }

  return true;
}
```