

Documentación del Proyecto: Juego de Disparo a Dianas con Láser con Arduino UNO

Descripción General

Este proyecto consiste en un juego interactivo donde el jugador debe disparar con un puntero láser a dianas que se levantan aleatoriamente mediante servomotores. Cada diana está equipada con un sensor de luz (LDR) que detecta el impacto del láser. El sistema está controlado por un Arduino UNO, incluye una pantalla LCD (compatibles 16x2, 16x4 o 20x4) con I2C, y dispone de botones de START y RESET para gestionar el juego.

Materiales Necesarios

Electrónica:

- 1x Arduino UNO
- 3x Fotorresistencias (LDR)
- 3x Resistencias de 10k Ω (para los divisores de voltaje)
- 3x Servomotores SG90 o similares
- 1x Pantalla LCD con interfaz I2C (puede ser de 16x2, 16x4 o 20x4)
- 1x Botón START (push button)
- 1x Botón RESET (push button)
- Cables Dupont
- Protoboard o soldadura para versión definitiva

Opcional:

- Carcasa de cartón o impresión 3D para soporte de las dianas
 - Puntero láser (tipo llavero)
-

Esquema de Conexiones

LDRs (x3):

Cada LDR se conecta como divisor resistivo:

- LDR entre +5V y pin analógico A0, A1, A2
- Resistencia de 10k Ω entre pin analógico y GND

Servos (x3):

- Señal: Pines digitales 3, 5 y 6

- VCC: +5V
- GND: GND

Botones:

- START: Pin digital 8 (pull-up interno activado)
- RESET: Pin digital 9 (pull-up interno activado)

LCD:

- Conexión I2C (SDA y SCL): A4 y A5 en Arduino UNO
- Compatible con pantallas de 16x2, 16x4 o 20x4 (ajustar número de columnas y filas en el código)

Código del Proyecto

El código está escrito en Arduino IDE y hace uso de las siguientes librerías:

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

Importante: Si usas una pantalla de 16x2 o 20x4, asegúrate de ajustar esta línea:

`LiquidCrystal_I2C lcd(0x27, 16, 4);` // Cambiar a `(0x27, 20, 4)` o `(0x27, 16, 2)` según el modelo usado

A continuación se muestra el código completo:

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

LiquidCrystal_I2C lcd(0x27, 20, 4);

const int numDianas = 3;
const int pinesLDR[numDianas] = {A0, A1, A2};
const int pinesServo[numDianas] = {3, 5, 6};
const int pinStart = 8;
const int pinReset = 9;

Servo servos[numDianas];

int puntuacion = 0;
int fallos = 0;
int ronda = 0;
bool juegoActivo = false;
unsigned long tiempoInicioRonda;
unsigned long tiempoLimiteRonda = 3000;
const int umbralLuz = 400;
```

```

bool botonStartPresionado = false;
bool botonResetPresionado = false;

void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Presiona START");

  pinMode(pinStart, INPUT_PULLUP);
  pinMode(pinReset, INPUT_PULLUP);

  for (int i = 0; i < numDianas; i++) {
    servos[i].attach(pinesServo[i]);
    servos[i].write(175);
  }
}

void loop() {
  if (!juegoActivo) {
    if (digitalRead(pinStart) == LOW && !botonStartPresionado) {
      botonStartPresionado = true;
      delay(100);
      iniciarJuego();
    }
    if (digitalRead(pinReset) == LOW && !botonResetPresionado) {
      botonResetPresionado = true;
      delay(100);
      reiniciarJuego();
    }
  } else {
    jugarRonda();
  }
}

void iniciarJuego() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Preparado...");
  delay(1000);
  for (int i = 3; i >= 1; i--) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Empieza en: ");
    lcd.print(i);
    delay(1000);
  }
  puntuacion = 0;
  fallos = 0;
}

```

```

    ronda = 1;
    juegoActivo = true;
    botonStartPresionado = false;
    tiempoLimiteRonda = 3000;
}

void jugarRonda() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Ronda ");
    lcd.print(ronda);
    lcd.setCursor(0, 1);
    lcd.print("Puntos: ");
    lcd.print(puntuacion);

    int dianaActiva = random(numDianas);
    servos[dianaActiva].write(90);
    tiempoInicioRonda = millis();

    bool impactoDetectado = false;
    while (millis() - tiempoInicioRonda < tiempoLimiteRonda) {
        if (analogRead(pinesLDR[dianaActiva]) > umbralLuz) {
            impactoDetectado = true;
            break;
        }
    }

    servos[dianaActiva].write(175);

    if (impactoDetectado) {
        puntuacion++;
    } else {
        fallos++;
    }

    delay(500);

    if (fallos >= 3) {
        mostrarResultados();
    } else {
        ronda++;
        if (tiempoLimiteRonda > 1000) tiempoLimiteRonda -= 200;
    }
}

void mostrarResultados() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Fin del juego");
}

```

```

    lcd.setCursor(0, 1);
    lcd.print("Puntos: ");
    lcd.print(puntuacion);
    lcd.setCursor(0, 2);
    lcd.print("Fallos: 3");
    juegoActivo = false;
}

void reiniciarJuego() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Reiniciando...");
    delay(1000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Presiona START");
    puntuacion = 0;
    fallos = 0;
    ronda = 0;
    tiempoLimiteRonda = 3000;
    juegoActivo = false;
    botonResetPresionado = false;
    botonStartPresionado = false;

    for (int i = 0; i < numDianas; i++) {
        servos[i].write(175);
    }
}

```



Funcionamiento del Juego

1. Al encender el sistema, la pantalla muestra: **“Presiona START para comenzar”**
2. El jugador pulsa START. El sistema realiza una cuenta atrás visible: **3... 2... 1...**
3. Se levanta aleatoriamente una diana. El jugador debe disparar con el láser.
4. Si el LDR detecta luz suficiente:
 - Se considera acierto
 - Se suma un punto
 - Se baja la diana
5. Si no hay impacto en el tiempo límite:
 - Se considera fallo
 - Se suma 1 fallo (hasta 3)
6. Cada ronda reduce el tiempo de reacción, aumentando la dificultad
7. Cuando se alcanzan 3 fallos, el juego termina y muestra el resultado
8. El sistema se reinicia sólo cuando se pulsa RESET

Ajustes y Personalización

- Puedes cambiar la dificultad inicial modificando el valor `tiempoLimiteRonda` en milisegundos
 - Ajusta el umbral de luz con `umbralLuz` según las condiciones de iluminación
 - Puedes modificar el código para levantar varias dianas simultáneamente
 - Si cambias el tamaño de la pantalla LCD, adapta las posiciones con `lcd.setCursor(col, fila)` según corresponda
-

Montaje Físico

- Utiliza soportes de cartón o impresión 3D para montar las dianas
 - Asegura que el LDR quede en el centro del blanco visible
 - Fija los servos de forma que al girar levanten la diana a 90°
-

Siguintes Pasos

- Mejoras: incluir sonidos, temporizador visual con LEDs, efectos de animación
 - Usar pantalla OLED o matriz LED para versiones compactas
 - Integrar puntuación en una app o envío por Bluetooth
-

Consideraciones Finales

Antes de cerrar tu montaje, revisa estos aspectos esenciales para asegurar un buen funcionamiento:

- 1. Revisar la posición 0, 90 y 180 de los servos:**
 - Asegúrate de que “posición bajada” y “posición levantada” estén correctamente configuradas. Puede variar según el modelo o montaje físico.
- 2. Revisar la tolerancia del umbral del sensor LDR:**
 - Ajusta el valor de `umbralLuz` observando el comportamiento real de los sensores con y sin láser.
- 3. Verificar la correspondencia entre motores y sensores:**
 - Asegúrate de que cada servo esté vinculado correctamente con su LDR correspondiente para evitar lecturas cruzadas o incoherencias.

Autores y Créditos

Taller de Introducción a Arduino - Juego de Disparo con Láser Organizado por:
EuskalEncounter 33 y Desarrollado por: Víctor Pardo Fecha: [25/07/2025]

✦ **Diviértete construyendo y personalizando tu propio juego de puntería láser con Arduino.**