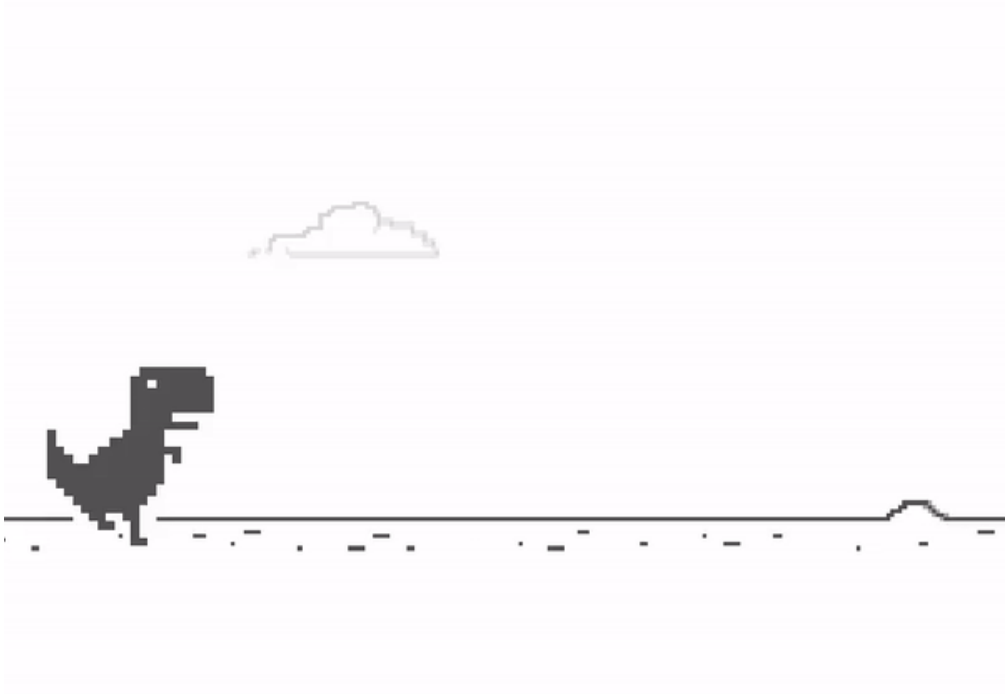
# Automated Chrome Dinosaur Game - Marinela Cotoi

## Introducere

Proiectul presupune automatizarea popularului jocului disponibil în browser-ul Google Chrome atunci când nu există conexiune la internet. Jocul este de tip “endless runner” și implică un dinozaur ce trebuie să se ferească de o serie de obstacole pentru a supraviețui si pentru a obtine puncte.

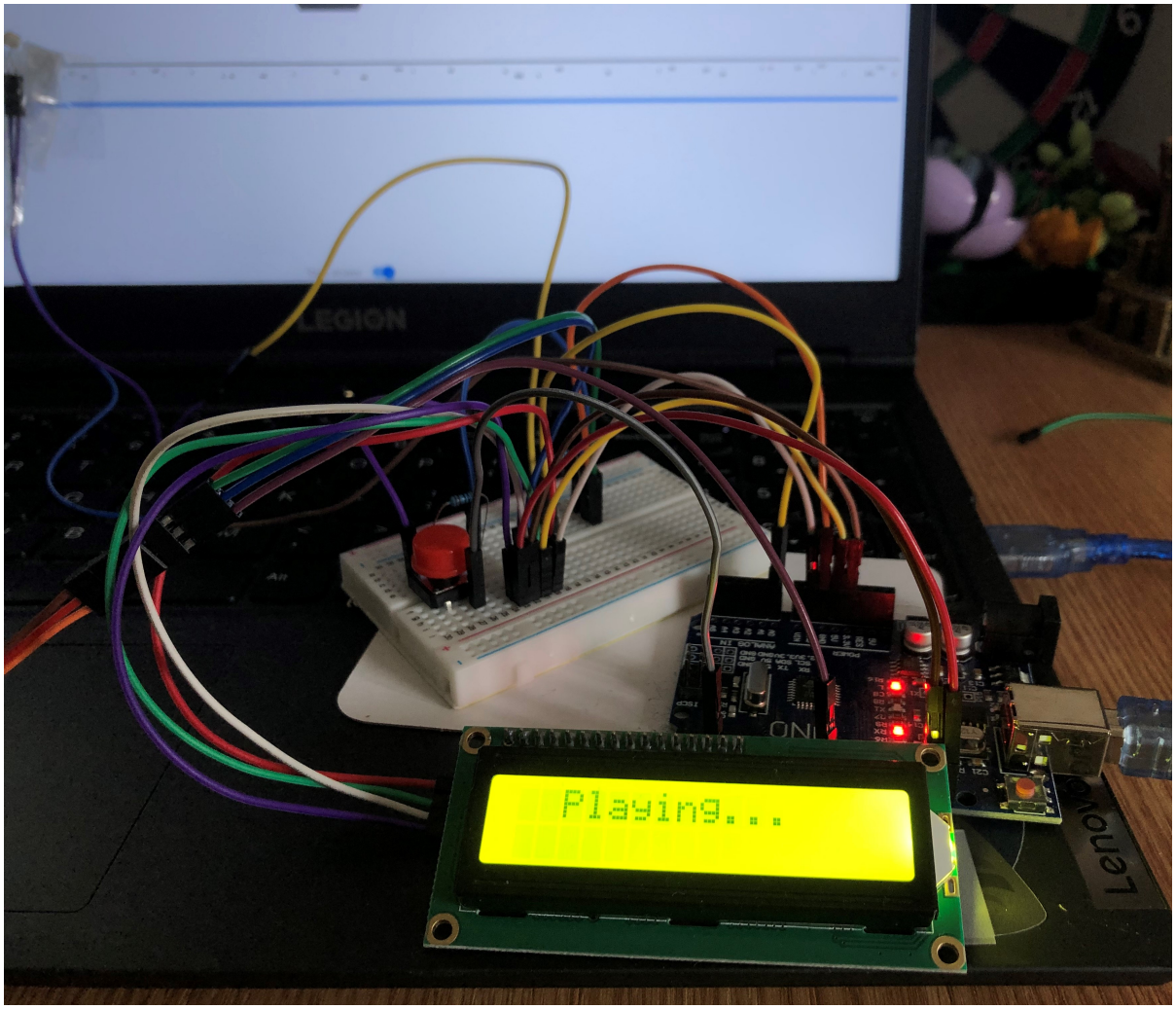


## Descriere generală

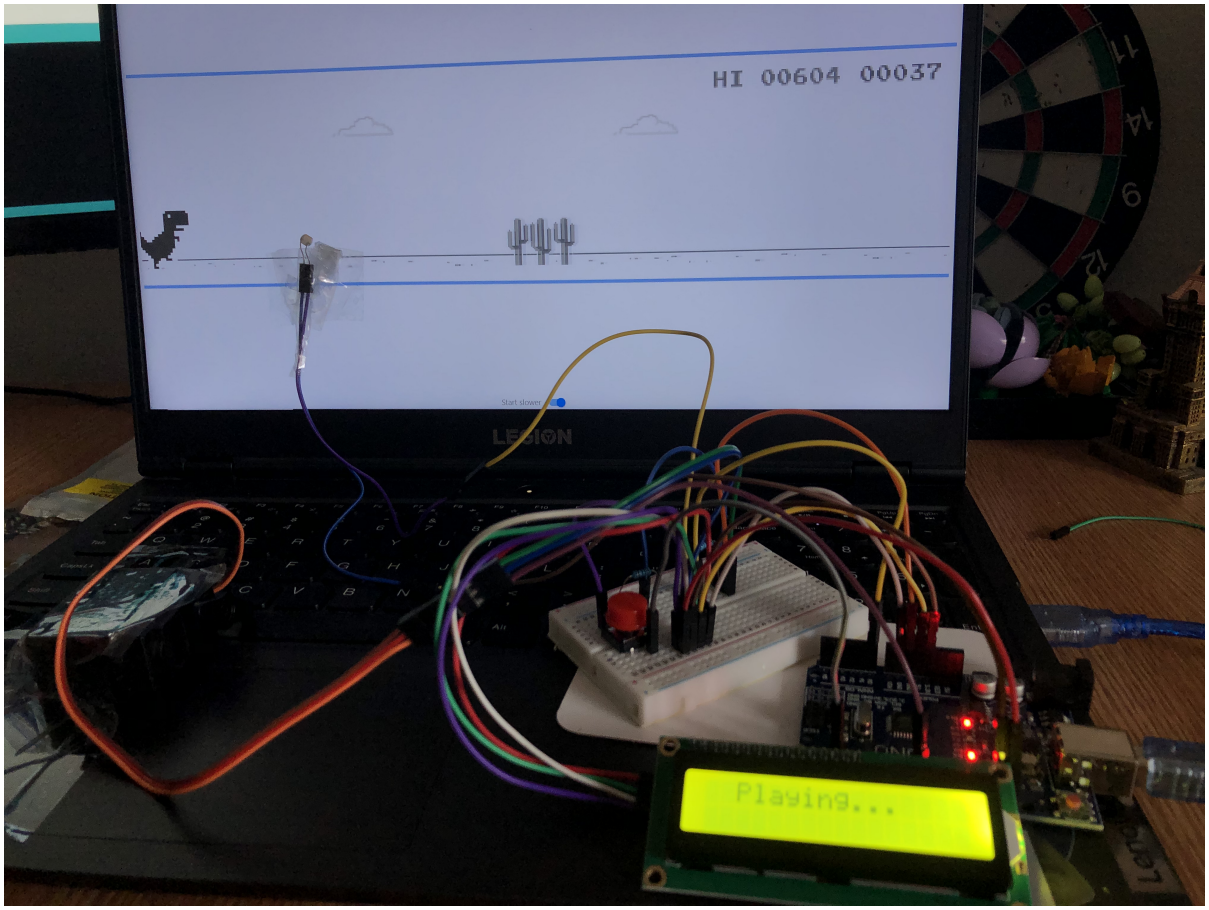
## Flow proiect

Inițial, ecranul LCD afișează mesajul **„Press the red button to start!”**, iar apăsarea butonului roșu determină ulterior prima mișcare a servo-motorului și respectiv, începutul jocului.

Ulterior, ecranul LCD afișează mesajul **„Playing..”**, iar jocul se desfășoară, prin utilizarea fotorezistorilor pentru a identifica obstacolele, respectiv momentele în care dinozaurul trebuie să sară.




Fotorezistorii vor fi lipiți la o distanță mică de dinozaur, pe ecranul laptopului. În momentul în care un obstacol se apropie, fotorezistorii vor detecta schimbarea luminii, și vor trimite semnalul către placa Arduino. Cum obstacolul este un copac, servomotorul va efectua o săritură apăsând butonul “space” de pe tastatură. Această acțiune va fi repetată în mod automat pe durata întregului joc, fără intervenția utilizatorului. Proiectul poate fi extins pentru a identifica și alte obstacole din joc și pentru a le evita automat.




## Schemă bloc

## Hardware design

## Componente hardware

Componentele necesare în dezvoltarea proiectului sunt următoarele:

- **Servomotor** - acesta va simula acțiunile unui jucător. Mai exact, va apăsa butonul 'space' al tastaturii laptopului în anumite momente ale jocului, atunci când se apropie un obstacol.

- **Fotorezistor** - pentru identificarea obstacolelor din joc

- **Ecran LCD - Afișaj electronic pe două rânduri** - pentru afișarea de mesaje în timpul jocului, precum și la începutul acestuia

- **Plăcuță Arduino Uno**
- **Breadboard**
- **Rezistențe, Fire, Buton**

## Schema electrică

## Software design

Pentru a simula începerea jocului la apăsarea butonului, am configurat o intrerupere, ce va opri

programul curent la sesizarea evenimentului.

```
ISR(PCINT2_vect) {
  /* Pin change type of interrupt */
  if (!(PIND & (1 << PIND2))) {
    buttonPressed = true;
  }
}

void setup() {
  ....
  /* Configure interrupt using registers */

  /* input */
  DDRD &= ~(1 << DDD2);
  /* input pullup */
  PORTD |= (1 << PORTD2);

  /* Enable the pin change interrupt*/
  PCMSK2 |= (1 << PCINT18);
  PCICR |= (1 << PCIE2);

  /* Enable interrupts */
  sei();
  ....
}
```

Pentru display-ul de mesaje folosind afișajul LCD pe două rânduri, am folosit biblioteca **LiquidCrystal_I2C**, inițializând lcd-ul astfel:

```
void setup() {
  ...
  /* Initialize LCD screen */
  lcd.init();
  /* Clear content on the screen */
  lcd.clear();
  /* Turn on backlight */
  lcd.backlight();

  /* Setup initial message on lcd */
  lcd.setCursor(2,0);
  lcd.print("Press the red");
  lcd.setCursor(0,1);
  lcd.print("button to start!");
  ...
}
```

Ultima bibliotecă folosită este biblioteca **Servo**, cu ajutorul căreia am gestionat miscările servomotorului.

```
void setup() {
```

```
  /* Initial servomotor configuration */

   /* Attach the servomotor to the object */
  servo.attach(servoPin);
  /* Set the servomotor position to the unpressed angle value */
  servo.write(unpress_angle);
}
```

Flow-ul programului final arată astfel:

```
void loop() {
  /* Print fotoresistor's value */
  Serial.println(analogRead(A0));
  Serial.println("\n");

  /* Check if the pushbutton is pressed. If it is, set the game
  as 'Playing' on the LCD screen*/
  if (buttonPressed && !playing) {
    playing = true;
    buttonPressed = false;

    /* Start the game */
    servo.write(press_angle);
    delay(100);

    /* Clear LCD and show text */
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("Playing.. ");
  }

  /* If the game has started */
  if (playing == true) {
    servo.write(unpress_angle);
    delay(10);

    /* If we catch an obstacle */
    if(analogRead(A0)< threshold)
    {
       servo.write(press_angle);
       /* Waits 100ms for the servo to reach the position */
       delay(100);
    }
  }
}
```

Fiecare apelare de funcție servo.write(), cu valoarea press_angle, este echivalentului apăsării fizice, în timp real, a butonului **Space** de pe tastatura laptopului, servo-motorul fiind lipit de acesta, declanșând astfel o săritură a dinozaurului în joc, după cum se poate vedea în video-ul atașat mai jos.

# Rezultate Obţinute

# Concluzii

Proiectul a reprezentat pentru mine o primă interacțiune independentă cu hardware-ul, cu piese fizice, în afara laboratorului și m-a ajutat să înțeleg cum să le configurez și cum interacționează unele cu altele, încorporând mai multe concepte învățate la PM în acest an. Mi s-a părut distractiv și nu foarte dificil să automatizez un joc real în această formă și datorită lui, am realizat câte lucruri faine putem face cu 2-3 componente și o placuță Arduino.

# Download

proiectpm2023.zip

# Bibliografie/Resurse


- https://ocw.cs.pub.ro/courses/pm/lab/lab2-2023
- https://docs.arduino.cc/built-in-examples/digital/Button
- https://docs.arduino.cc/learn/electronics/servo-motors
- https://www.instructables.com/How-to-use-a-photoresistor-or-photocell-Arduino-Tu/


Export to PDF

From:
http://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
**http://ocw.cs.pub.ro/courses/pm/prj2023/vstoica/marinela-cotoi**

Last update: **2023/05/29 08:47**