

Dino Game

Introducere

Acest proiect are rolul de a mima celebrul joc offline "Chrome Dino", dezvoltat de Google. În acest proiect, jucătorul ghidează un personaj pixelat printr-un peisaj cu derulare laterală, evitând obstacolele pentru a obține un scor mai mare.

Descriere generală

- Jocul va începe odată ce apeși butonul.
- Scopul tău este să protejezi persoana de blocurile care o lovesc.
- Scorul va fi afișat în partea dreaptă sus.

Astfel, utilizatorul va apăsa butonul ori de câte ori dorește ca personajul să efectueze o saritură. Saritura se realizează în momentul întâlnirii unui obstacol, înainte de potențialul contact cu acesta.

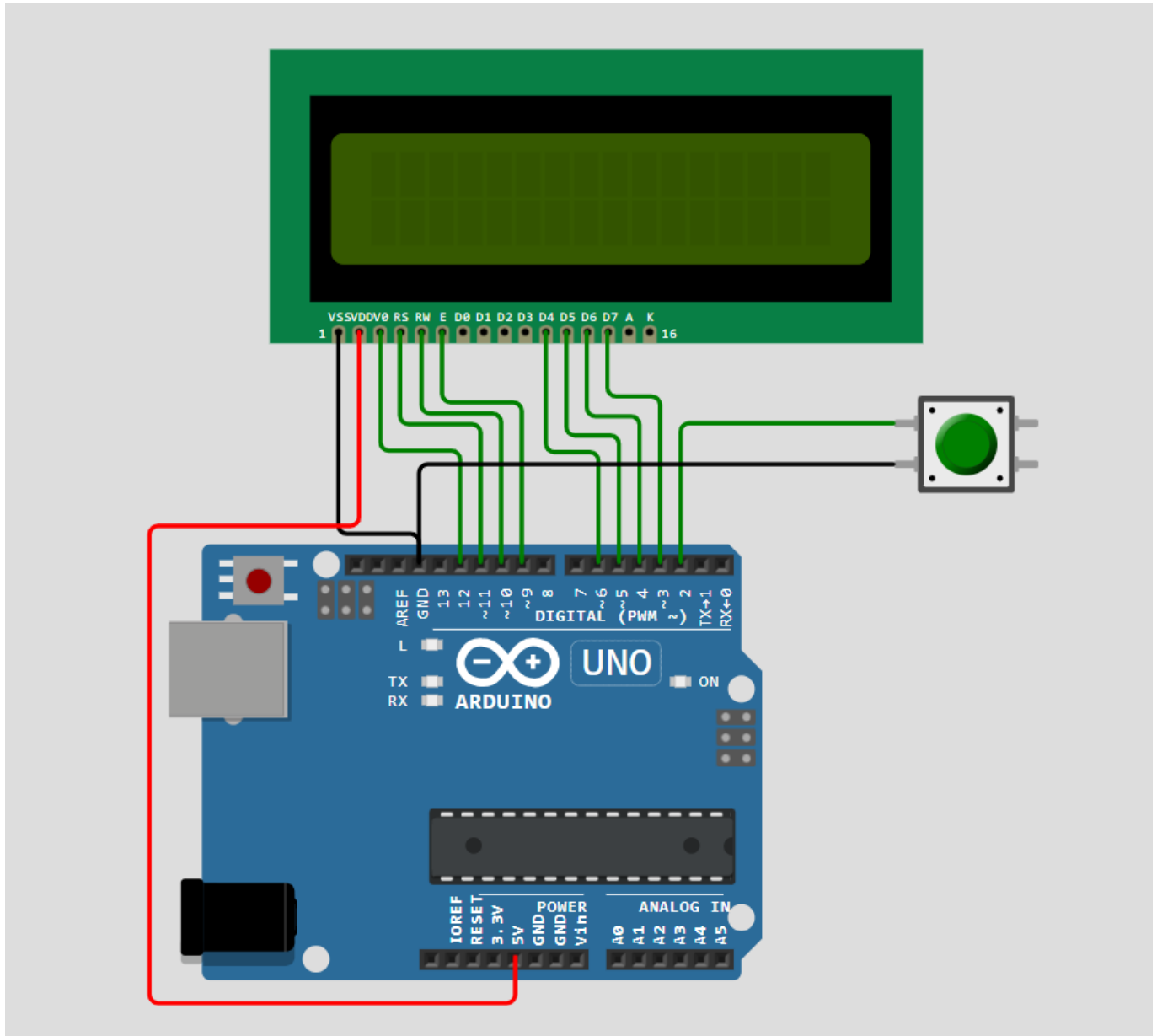


Hardware Design

Listă de piese:

- Arduino UNO
- Fire de lagatura
- Breadboard
- Switch
- LCD 1602
- Cablu USB pentru alimentarea UNO

Schema electrică:



Pini utilizați: Arduino Uno:

- GND (GND.1): Conectat la pinul VSS al LCD-ului pentru a furniza legătură la masă (GND).
- 5V: Conectat la pinul VDD al LCD-ului pentru a furniza alimentare la 5V.
- Pin 9 (E): Conectat la pinul E (Enable) al LCD-ului pentru controlul modului de execuție.
- Pin 10 (RW): Conectat la pinul RW (Read/Write) al LCD-ului pentru controlul direcției de scriere.
- Pin 11 (RS): Conectat la pinul RS (Register Select) al LCD-ului pentru a selecta între date și instrucțiuni.
- Pin 12 (V0): Conectat la pinul V0 al LCD-ului pentru a controla contrastul afișajului.
- Pin 6 (D4): Conectat la pinul D4 al LCD-ului pentru comunicația în modul 4-bit.
- Pin 5 (D5): Conectat la pinul D5 al LCD-ului pentru comunicația în modul 4-bit.
- Pin 4 (D6): Conectat la pinul D6 al LCD-ului pentru comunicația în modul 4-bit.
- Pin 3 (D7): Conectat la pinul D7 al LCD-ului pentru comunicația în modul 4-bit.
- Pin 2: Conectat la butonul pushbutton pentru a detecta apăsarea butonului.

LCD 1602:

- VSS: Conectat la GND al Arduino Uno pentru a furniza legătură la masă (GND).
- VDD: Conectat la 5V al Arduino Uno pentru a furniza alimentare la 5V.

- E (Enable): Conectat la pinul 9 al Arduino Uno pentru controlul modului de execuție.
- RW (Read/Write): Conectat la pinul 10 al Arduino Uno pentru controlul direcției de scriere.
- RS (Register Select): Conectat la pinul 11 al Arduino Uno pentru a selecta între date și instrucțiuni.
- V0: Conectat la pinul 12 al Arduino Uno pentru a controla contrastul afișajului.
- D4, D5, D6, D7: Conectate la pinii 6, 5, 4 și 3 ai Arduino Uno pentru comunicația în modul 4-bit.

Buton:

- 1.I: Conectat la pinul 2 al Arduino Uno pentru a detecta apăsarea butonului.
- 2.I: Conectat la GND al Arduino Uno pentru a furniza legătură la masă (GND).

Software Design

Mediu de Dezvoltare: Codul aplicației este dezvoltat folosind Arduino IDE.

Librării și Surse 3rd-Party: Codul utilizează biblioteca standard LiquidCrystal.h, care facilitează interacțiunea cu afișajele LCD bazate pe controlere HD44780. Aceasta permite controlul ecranului pentru a afișa caractere și grafică personalizată.

Algoritmi și Structuri Implementate: Aplicația implementează un joc simplu de tip runner, unde jucătorul controlează un erou ce se deplasează printr-un teren care se derulează continuu de la dreapta la stânga. Principalele componente ale algoritmului includ:

- Inițializarea Graficii: Funcția initializeGraphics() definește și încarcă caractere personalizate pentru afișajul LCD, utilizând funcția lcd.createChar() pentru a crea sprite-uri pentru erou și teren.
- Avansarea Terenului: Funcția advanceTerrain() deplasează terenul cu o unitate spre stânga, ajustând caracterele pentru a simula mișcarea continuă a terenului.
- Desenarea Eroului: Funcția drawHero() gestionează afișarea eroului în diferite poziții (alergare, săritură) și detectează coliziunile cu terenul.
- Gestionarea Interacțiunilor: Funcția buttonPush() tratează apăsările de buton folosind întreruperi hardware pentru a declanșa salturi ale eroului. În funcția loop(), starea jocului este actualizată continuu, avansând terenul, schimbând poziția eroului și verificând coliziunile. Jocul reîncepe atunci când este detectată o coliziune.
- Controlul Jocului: Folosind variabile statice pentru a menține starea jocului între iterațiile buclei principale, jocul este controlat printr-o serie de stări și tranziții care determină acțiunile eroului și actualizarea ecranului.

Surse și Funcții Implementate:

- initializeGraphics(): Definește caracterele personalizate pentru erou și teren și inițializează matricele de teren.
- advanceTerrain(char terrain, byte newTerrain) : Actualizează matricele de teren prin deplasarea conținutului spre stânga și adăugarea de noi elemente la capătul din dreapta.
- drawHero(byte position, char terrainUpper, char terrainLower, unsigned int score): Desenează eroul în poziția sa actuală și verifică coliziunile cu terenul. Afișează și scorul curent pe ecranul LCD.
- buttonPush(): Setează un flag atunci când butonul este apăsat, utilizat pentru a declanșa saltul eroului.
- setup(): Configurarea inițială a pinilor, afișajului LCD și întreruperilor hardware.
- loop(): Buclează continuu pentru a actualiza starea jocului, incluzând avansarea terenului,

gestionarea poziției eroului și verificarea coliziunilor. De asemenea, gestionează starea de așteptare înainte de începerea unui nou joc.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/rpascale/darius.calota>



Last update: **2024/05/21 21:33**