A black background with red lines

Description automatically generated

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**SPECIALIZAREA CALCULATOARE**

**Documentație Dino Game**

Disciplina: Proiectare cu microprocesoare (PMP)

Student: Pojar Andrei-Gabriel

An: III

Grupa: 30233

Profesor coordonator: Șopterean Andrei

**CUPRINS**

1. Introducere

* Motivație
* Obiective

1. Descrierea soluției
2. Domeniu de aplicare
3. Implementare
4. Avantaje
5. Schema de montaj
6. Bibliografie
7. **Introducere**

Proiectul ‚,Dino Game’’ reprezintă o încercare captivantă de a aduce universul jocurilor pe platforma Arduino. Inspirat de popularul joc offline ,,Chrome Dino Run’’, acest proiect aduce aceeași experiență distractivă și captivantă pe o platformă hardware accesibilă, precum Arduino.

* **Motivație**

Proiectul a fost conceput ca o modalitate de introducere în programarea hardware și dezvoltarea de jocuri simple pe platforma Arduino. Motivația principală este să ofere un mediu practic și interactiv pentru începători să exploreze programarea embedded și să dobândească abilități în dezvoltarea de proiecte de tip joc.

* **Obiective**

Am stabilit următoarele obiective pentru proiectul meu:

* Furnizarea unei platforme educaționale pentru învățarea programării Arduino
* Crearea unui joc care să implice utilizatorul în procesul de control al unui personaj într-un mediu virtual
* Încurajarea dezvoltatorilor amatori să exploreze potențialul programării hardware pentru a crea experiențe interactive

1. **Descrierea soluției**

Proiectul implementează un joc simplu pe o placă Arduino, utilizând un ecran LCD pentru afișarea graficelor și a stării jocului și diferite componente electronice . Un dinozaur controlat de un utilizator trebuie să evite obstacolele care pot fi ori cactuși ori păsări pentru a obține un scor cât mai mare. Jocul oferă o introducere practică în concept pentru gestionarea timpului, manipularea LCD-ului și gestionarea intrărilor utilizatorului.

1. **Domeniul de aplicare**

Proiectul meu are un domeniu de aplicare extins, fiind potrivit pentru:

* Cursuri de programare în Arduino
* Activități educaționale în școli sau în cadrul comunității pentru a stimula interesul pentru programarea hardware
* Proiecte DIY: Explorarea și adaptarea proiectului ,,Dino Game’’ pentru a crea variante personalizate
* Interacțiunea elevilor cu diferite dispozitive electronice ce au fost folosite în întocmirea acestui joc cum ar fi: buzzer, ecran LCD, placa Arduino Uno, etc.

1. **Implementare**

Proiectul utilizează o serie de componente hardware comune cum ar fi un ecran LCD, o placă Arduino Uno, un buzzer, fire pentru conexiune, un breadboard, un potențiometru și rezistoare. Codul sursă include gestionarea graficelor pentru personaje și obstacole, logica de joc și interacțiunea cu utilizatorul prin intermediul butoanelor.

Mai întâi am declarat și inițializat toate bibliotecile și variabilele de care am nevoie:

A screenshot of a computer program

Description automatically generated

Mai apoi am funcția setup() care setează pinii 2-17 ca ieșiri și pinii butoanelor respectiv al buzzer-ului ca intrări, respectiv ieșiri. Tot în această funcție se inițializează LCD-ul și se definesc caracterele personalizate, iar apoi se afișează un mesaj de bun venit pe LCD la începutul jocului.

A screenshot of a computer program

Description automatically generated

Apoi am funcția loop() care verifică dacă butonul de restart a fost apăsat și în caz că a fost apăsat se resetează jocul. Dacă jocul nu s-a încheiat se apelează funcția Game\_Code().

A screenshot of a computer program

Description automatically generated

Mai departe am realizat o funcție care gestionează atunci când jucătorul a pierdut. Acestă funcție se numește game\_over(). Verific daca scorul curent depășește scorul maxim și dacă depășește afișez și la scorul curent și la high score aceeași variabilă deoarece cel mai mare scor se updatează la scorul care a bătut cel mai mare scor anterior. Dacă scorul curent nu depășește high score atunci voi afișa cel mai mare scor înregistrat+ scorul curent obținut. De asemenea în partea de jos a ecranului se afișează mesajul de GAME OVER.

A screenshot of a computer program

Description automatically generated

Mai departe vom discuta despre funcția care practic realizează acest joc și anume Game\_Code(). Mai întâi actualizăm starea dinozaurului la intervale regulate. Această condiție asigură că starea dinozaurului(reprezentată de variabila flag) este actualizată la intervalele specificate de variabila period. Flag practic reprezintă dacă dinozaurul merge normal sau ridică un picior.

A screen shot of a computer program

Description automatically generated

Apoi actualizăm poziția obstacolului la intervale regulate verificând dacă a trecut intervalul branchPeriod de la ultima actualizare a poziției obstacolului. Daca branchColumn devine mai mic ca 0 atunci actualizez branchColumn înapoi la 17. De obicei atunci când un obstacol iese din partea stângă a ecranului vrem să îl mutăm înapoi în partea dreaptă pentru a reda impresia că se reîntoarce și de aceea resetăm branchColumn(coloana obstacolului) înapoi la 17. Se modifică și perioada și tipul de obstacol cu ajutorul funcției random. Pe măsură ce jocul avansează jocul este făcut mai dificil și de aceea branchPeriod este redus cu valoarea variabilei acceleration de fiecare dată când un obstacol trece de-a lungul ecranului. Prin scăderea acestui interval obstacolele apar mai repede. Mai departe se asigură că obstacolul nu iese din limitele ecranului și setăm variabila a la 1 pentru a indica că este necesară afișarea unui nou obstacol.

A screenshot of a computer program

Description automatically generated

Apoi actualizăm cursorul pentru ultimele doua linii, adică mai precis afișarea dinozaurului în funcție de starea acestuia. Verificăm mai întâi dacă este nevoie de afișarea dinozaurului și dacă dinozaurul este în stare de mișcare se setează cursorul la poziția dinozaurului și afișează în mod corespunzător dinozaurul în funcție de stare( mișcare sau sare).

A screenshot of a computer code

Description automatically generated

Următorul pas este actualizarea obstacolului. Acest lucru se face verificând mai înâi dacă este necesară afișarea unui nou obstacol ( if (a==1) ). În funcție de valoarea dată de randomNumber se alege să se afișeze un cactus(una din cele două stări definite anterior) sau o pasăre. Dacă obstacolul este o pasăre se asigură că pasărea este afișată corect pe ecran și resetăm flagul de necesitate a afișării unui nou obstacol la 0.

A screenshot of a computer program

Description automatically generated

Mai departe în această funcție este să verificăm butonul de săritură. Dacă butonul de săritură este apăsat atunci vom seta variabilele pentru a simula săritura acestuia. Dinozaurul poate fi în stare de mișcare sau în săritură. Dacă butonul de jump este apăsat atunci se va emite un sunet prin buzzer, în caz contrar se resetează variabilele b și c care sunt utilizate pentru a ajusta afișarea obstacolelor pe ecran și a variabilei d .

A screenshot of a computer program

Description automatically generated

În cele ce urmează, în funcția Game\_Code() va trebui să actualizez scorul maxim și să afișez scorul. Mai întâi actualizez high score-ul dacă scorul actual este mai mare decât high score-ul actual. Apoi voi verifica dacă a trecut perioada scorePeriod de la ultima actualizare a codului și actualizez scoreTimer și incrementez scorul. Apoi voi face o verificare pentru nivelul jocului și atunci când este atins un alt nivel se vor emite două tonuri cu ajutorul buzzer-ului. Aceste două tonuri se vor emite pentru fiecare prag de 100 de puncte atins. Mai apoi actualizăm scorul cu valoarea scorului curent și afișăm scorurile pe ecran.

A screenshot of a computer program

Description automatically generated

În continuare ne ocupăm de săritura dinozaurului, adică când a revenit din săritură să ștergem caracterele care au apărut când a avut loc săritura. Aceasta are loc numai dacă starea butonului de jump este diferită față de cea anterioară. Mai apoi vom verifica când pierdem jocul. Mai precis pierdem jocul atunci când butonul de jump e apăsat și apare o coliziune la săritură sau când efectiv intrăm într-un obstacol fără să sărim. Când pierdem jocul se vor emite sunete cu ajutorul buzzer-ului și se va apela funcția game\_over().

A screenshot of a computer program

Description automatically generated

1. **Avantaje**

* Design simplu și interfață utilizator prietenoasă
* Extensibilitate și adaptabilitate
* Utilizarea efectelor sonore pentru feedback
* Integrarea cu ecran LCD și butoane
* Compatibilitate cu diverse platforme Arduino

1. **Schema de montaj**

A circuit board with a screen and wires

Description automatically generated

Realizată în programul Cirkit Designer.

1. **Bibliografie**

* <https://uk.beta-layout.com/download/rk/RK-10290_410.pdf>
* <https://docs.arduino.cc/learn/electronics/lcd-displays>
* <https://docs.arduino.cc/static/7d7b6e99f40c7e55f2e9c6175c6db5b5/260cd/LCD_Base_bb_Fritz.png>
* <https://www.cirkitstudio.com/index.html>
* <https://projecthub.arduino.cc/Unsigned_Arduino/the-chrome-dino-game-on-an-lcd-shield-8520e3>