

Arcade Mascot Game

Introducere

Nume: Ionescu Stefan

Grupa: 331CD

Arcade mascot game este un joc de tip arcade in care mascota ocoleste obstacole obtinand un anumit scor

Proiectul propune o experiență de joc care îmbină distracția și relaxarea cu provocările și excitarea. Jucătorul este invitat să se bucure de o atmosferă plină de culoare și creativitate, care să îi ofere o pauză plăcută în rutina zilnică. În același timp, elementele de provocare și dificultate sunt integrate strategic în joc pentru a menține interesul și a oferi o senzație de satisfacție atunci când acestea sunt depășite.

Ideea de baza al acestui proiect a constat in oferirea unei modalitati rapide de divertisment accesibila pentru toti

Cu toate că jocurile digitale sunt mult mai populare și căutate în ziua de azi, jocurile de tip arcade au un farmec aparte și o istorie bogată, oferind o experiență de joc simplă și captivantă. Ele aduc nostalgia pentru perioada în care arcade-urile erau destinația principală pentru întâlnirile cu prietenii sau pentru competițiile amicale, iar gameplay-ul lor rapid și accesibil continuă să atragă atât jucătorii de ocazie, cât și pe cei pasionați de retro-gaming.

Descriere generală

Schema bloc:



Hardware Design

Lista componente:

- o placuta Arduino Uno
- un ecran LCD 1602 cu interfata I2C
- butoane
- un breadboard
- fire
- 4 rezistente

- un buzzer
- fire mama-tata pentru conectarea LCD-ului la placa Arduino
- fire tata-tata pentru conectarea componentelor(buzzer, rezistente, LED, butoanele) la placa Arduino
- trei led-uri

Schema electrica:



Afisare LCD:



Am afisat mesajul Press Start pe ecranul LCD



Utilizarea componentelor:

- Ecran LCD→ Afisarea miscarilor mascotei si a obstacolelor (Pini folositi : SDA-A4(Arduino), SCL-A5(Arduino), VCC-5V(Arduino), GND-GND(Arduino))
- Butoane→ deplasarea mascotei si punerea in pauza a jocului (Pini folositi : Buton1-Pin7(Arduino) Buton1-GND, Buton2-Pin2(Arduino) Buton2-GND)
- LED-uri→ daca jucatorul va castiga se va aprinde led-ul verde, daca va pierde se va aprinde led-ul rosu si daca se pune pauza jocului se va aprinde led-ul galben (LED rosu-Pin11(Arduino) LED rosu-GND; LED verde-Pin13(Arduino) LED verde-GND; LED galben-Pin9(Arduino) LED galben-GND)
- Buzzer→ genereaza sunete (Am pus rezistenta la buzzer pentru reducerea volumul sunetului)
- Rezistentele sunt toate de 10K(ohmi).
- LED-urile sunt insotite de rezistente din motive de protectie impotriva unui flux de instensitate electrica prea mare

Software Design

In cadrul acestui proiect, actiunea se petrece in interiorul functiei de loop. Daca nu ruleaza jocul si nu este pauza atunci verific daca s-a scurs counter-ul care este timpul alocat jucatorului in acest caz 20 de secunde. Daca da, atunci inseamna ca nu a avut loc nici o coliziune si jucatorul a castigat. In continuare, resetez numaratoarea si jocul o va lua de la capat cu mesajul "BEGIN ?" "PRESS START !"; tind sa mentionez faptul ca in aceasta faza, este si muzica ce va fi rulata am ales PinkPanther; muzica va fi redata pe masura ce mesajele "BEGIN ?" "PRESS START !" sunt afisate. In acest ipostaz, jucatorul se afla in postura de repaus. Ca sa inceapa jocul apas pe buton. Pentru realizarea unei astfel de operatii ma folosesc de intreruperi. Daca se va apasa pe buton, jocul incepe si harta incepe sa fie desenata. Aceasta contine mai multe obstacole sub forma unor copaci. Fiecare copac are un timp de viata cat este afisat pe ecran. Jucatorul in sine va fi mereu pozitionat pe celula de jos. Pentru a realiza lucrurile mai dinamic, am inclus si niste miscari de sarituri pentru jucator in matricea de miscare. Felul cum harta va fi generata in continuare este realizata tot in loop, unde in functie de pozitia jucatorului decid cum va arata urmatorul frame. Am si un buton de pause , care are rolul de a pune pauza jocului oricand jucatorul are nevoie iar dupa cateva secunde de la apasarea acestui buton se va rezuma la joc. O data ce butonul de pauza a fost apasat, un mesaj : "GAME PAUSED" va fi afisat pe ecranul LCD. Inotit de acest mesaj este printat si scorul actual al jucatorului. Daca a avut loc o coliziune, LED-ul rosu se va aprinde alaturi de mesajul "LOST ! AGAIN ?" urmat de scorul obtinut; daca aceste doua mesaje cu "GAME PAUSED" si "LOST ! AGAIN ?" sunt afisate pe ecran o muzica va fi redata este muzica de GameOver in cod. Daca jocul va fi pus pe pauza LED-ul galben se va aprinde, alaturi de mesajul "GAME PAUSED" si scorul actual. Iar daca jucatorul va castiga se va afisa mesajul "YOU WON" "ANOTHER ONE ?", simultan cu el LED-ul verde se va aprinde.

In functia `map_initialisation`, harta este creata, iar in functia `map_animation` se va decide care va fi urmatorul pas in reprezentarea hartii.

In functia `player_design`, decid urmatoarele miscari ale playerului in functie de pozitia actuala. Tot aici in partea dreapta sus a ecranului reprezinta timpul scurs.

In setup se fac initializarile pentru butoane, registrii folositi pentru intreruperi atat pentru butoane cat si pentru timer. Initializarea pentru buzzer este de asemenea reprezentata aici. In ceea ce priveste intreruperile si registrii pentru ele au fost realizate in stilul laboratorului. Pentru rutinele de intreruperi am folosit ISR.

In functia `GAME_OVER_MELODY` redau melodia de gameover in caz de pauza sau de pierdere.

Descrierea mediului de dezvoltare:

- mediu de dezvoltare : ARDUINO IDE
- LIBRARII :
 - `Wire.h` → folosita pentru lucrul cu I2C, specific comunicarea intre I2C si componente ARDUINO
 - `"LiquidCrystal_I2C.h"` → folosita pentru controlul ecranului LCD prin intermediul I2C. Pentru initializarea controlul si afisarea de informatii pe ecranul LCD.
 - `"LCD.h"` → folosita pentru afisarea mesajelor pe LCD si controlul lor.

Laboratoare Incluse

- Laborator 0 : GPIO → folosit pentru ajutorul utilizarii placutei ARDUINO
- Laborator 2 : Intreruperi → folosite intreruperile pentru butoane si timer
- Laborator 3: Timere → folosit un timer pentru timpul de joc
- Laborator 6: I2C → Am folosit un adaptor I2C pentru LCD si utilizat biblioteci I2C pentru gestionarea si controlul mesajelor pe ecranul LCD

Rezultate Obținute



Download

- Archive : [arcade_mascot_game_1.zip](#)
- GitHub : <https://github.com/stefanione/Arcade-Mascot-Game>

Jurnal

- 21 aprilie → inceperea realizarii paginii de wiki
- 15 mai → inceperea montajului proiectului
- 17 mai → finalizarea montajului
- 19 mai → inceperea realizarii implementarii Software
- 23 mai → finalizarea realizarii software
- 25 mai → definitivarea paginii de wiki

Bibliografie/Resurse

- <https://maxpromer.github.io/LCD-Character-Creator/>
- <https://www.circuito.io/blog/arduino-uno-pinout/>
- <https://github.com/robsoncouto/arduino-songs/tree/master/pinkpanther>
- <https://www.youtube.com/watch?v=CvqHkXeXN3M&t=74s>
- <https://www.youtube.com/watch?v=EAeuxjtkumM>
- <https://www.youtube.com/watch?v=Z0Dj1Q4fFJE&t=107s>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab2-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2023>

[Export to PDF](#)

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

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2024/cpatru/stefan.ionescu1609>



Last update: **2024/05/27 10:36**