

2048 Game

Introducere

Proiectul isi propune implementarea jocului 2048. Initial pornim cu un grid 4x4 cu 2 valori intre 2 si 4 in casute aleatoare. Exista 4 tipuri de mutari: sus, jos, stanga, dreapta. Scopul jocului este sa atingi valoarea 2048

Descriere generală

În jocul 2048, există patru tipuri de mutări disponibile: sus, jos, stânga și dreapta, iar fiecare mutare implică deplasarea tuturor pieselor în direcția aleasă. Piesele cu aceeași valoare se combină și se eliberează unul dintre câmpuri. De asemenea, la fiecare mișcare, o valoare de 2 sau 4 este adăugată aleatoriu într-un câmp gol din grilă. Mutarile se vor realiza cu ajutorul unui joystick. Jocul se încheie atunci când nu mai sunt mutări disponibile (când grila este plină) sau atunci când se obține piesa cu valoarea 2048.

Schema bloc



Hardware Design

- Arduino UNO
- Display OLED 128x64 - 0.96" SPI
- Joystick
- Breadboard



Software Design

Codul a fost scris folosind Arduino IDE. Pentru comunicarea cu display-ul OLED am folosit biblioteca u8glib. Jocul poate primi 4 inputuri din partea joystick-ului : sus, jos, stanga, dreapta. Am implementat

un delay astfel incat atunci cand maneta revine la pozitia initiala sa nu se inregistreze niciun fel de input.

Codul este impartit in 3 parti:

Secțiunea 1 - Implementarea jocului 2048:

Matricea `mat` reprezintă tabla de joc 4×4 , inițializată cu zero. Funcția `add_new_2()` adaugă o valoare 2 într-o poziție liberă aleatoare pe tabla de joc. Funcția `get_current_state()` verifică starea curentă a jocului și returnează: 2 dacă unul dintre elementele matricei are valoarea 2048 (adică jucătorul a câștigat). 1 dacă există cel puțin o poziție liberă pe tabla de joc. 0 dacă jocul poate continua. -1 dacă jucătorul a pierdut. Funcțiile `compress()`, `merge()`, `reverse()`, `transpose()` și `move_left()`, `move_right()`, `move_up()`, `move_down()` sunt utilizate pentru a efectua mișcările jucătorului în joc. Acestea rearanjează valorile matricei în funcție de mișcarea specificată și combină valorile identice. Funcția `reset_matrix()` resetează tabla de joc la starea inițială. Funcția `print_matrix()` afișează tabla de joc în Serial Monitor (această funcție nu este utilizată în codul principal al jocului).

Secțiunea 2 - Implementarea afișării pe ecran:

Biblioteca `U8glib` este utilizată pentru a controla ecranul SSD1306 de tip OLED, cu rezoluție 128×64 de pixeli. Funcția `draw()` este apelată pentru a desena starea curentă a jocului pe ecran. Dacă jucătorul a câștigat sau a pierdut, se va afișa un mesaj corespunzător. Funcția `setup_display()` este utilizată pentru a configura ecranul la pornire. Funcțiile `draw_win()` și `draw_lose()` desenează mesajele corespunzătoare pe ecran atunci când jucătorul a câștigat sau a pierdut. Funcția `draw_grid()` desenează tabla de joc și valorile matricei pe ecran.

Secțiunea 3 - Implementarea citirii de la joystick:

Variabilele și constantele din această secțiune sunt utilizate pentru citirea valorilor de la joystick. În funcția `setup()`, se configurează pinii joystick-ului și se inițializează matricea de joc cu o valoare 2 într-o casuta random. În funcția `game_loop()`, se citesc valorile de pe joystick și se realizează mișcările în funcție de direcția deplasării indicate de joystick. După fiecare mișcare, se verifică starea jocului utilizând funcția `get_current_state()`. Dacă jocul continuă, se adaugă o valoare 2 într-o poziție liberă și se afișează starea jocului pe ecran. Dacă jucătorul a câștigat sau a pierdut, se afișează mesajul corespunzător pe ecran. Funcția `loop()` este responsabilă pentru apelarea în buclă a funcțiilor `game_loop()` și `draw()`, pentru a actualiza starea jocului și afișarea pe ecran.

Rezultate Obținute

Am realizat implementarea jocului 2048. Proiectul permite jucătorului să controleze mișcările utilizând un joystick, iar starea jocului este afișată pe un ecran OLED. Am realizat exact tot ceea ce am dorit să fac pentru acest proiect.

Concluzii

În urma implementării acestui proiect, am învățat mai multe lucruri importante atât pe partea de

software, cât și pe partea de hardware. Pe partea de software, am creat funcții și module clare și corespunzătoare pentru diversele sarcini ale proiectului, ceea ce a facilitat gestionarea și dezvoltarea ulterioară a codului. Am dobândit, de asemenea, cunoștințe despre manipularea matricelor și algoritmi de joc. Pe partea de hardware, am învățat cum să conectez și să configurez corect componentele necesare. Am realizat conexiunile adecvate între joystick și placa Arduino, precum și între ecranul OLED și placa Arduino. Am înțeles cum să configurez pinii și să asigur o alimentare corespunzătoare pentru componentele hardware.

Am utilizat biblioteca U8glib pentru afișajul pe ecranul OLED. Prin studierea documentației acestei biblioteci, am învățat cum să controlez și să utilizez funcțiile și metodele oferite pentru a desena grafice, text și alte elemente pe ecranul OLED. Astfel, am creat o interfață utilizator grafică, care a permis interacțiunea intuitivă cu jocul.

În concluzie, prin implementarea acestui proiect, am dobândit abilități practice în dezvoltarea de software și interacțiunea cu componente hardware. Am învățat cum să integrez componente hardware într-un proiect, să lucrez cu matrici și algoritmi de joc, precum și să realizez o interfață utilizator grafică.

Download

[ghioceI_andrei_332cc.zip](#)

Jurnal

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

Bibliografie/Resurse

Biblioteca u8glib - folosită pentru afișarea pe display : <https://github.com/olikraus/u8glib>

Display OLED :

https://ardushop.ro/ro/home/2647-display-oled-128x64.html?gclid=Cj0KCQjw3a2iBhCFARIsAD4jQB1HnTrGmXnY36S_vgQN8TI2yhgnnf_pi1Fxd6pW38BFpCnTYAkXm2YaAtCOEALw_wcB

Datasheet joystick : http://www.energiazero.org/arduino_sensori/joystick_module.pdf

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/alucaci/andreighiocol>



Last update: **2023/05/29 21:11**