**UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA**

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**MATERIE: PROIECTARE CU MICROPROCESOARE**

**PROIECT : SNAKE GAME**

**STUDENT : VATAMANU FLORINA-MIHAELA**

**AN III, GRUPA 30326**

Ideea proiectului: jucatorul controleaza o linie(care reprezinta sarpele,este formata din 6 pixeli in cazul proiectului meu); scopul este acela de a creste sarpele in lungime,in timp ce inainteaza controlat de butoane ,sarpele in sine fiind obstacolul principal.

Jucatorul pierde atunci cand sarpele se atinge pe sine(head==tail).

Pentru realizarea acestui proiect ,Snake Game, am folosit placuta Arduino Uno R3, LCD display 16x2, breadboard,potentiometru si 4 butoane. Proiectul este realizat in simulatorul tinkercad.

Diagram

Description automatically generated

Pentru inceput , am folosit libraria <LiquidCrystal.h>, care mi-a permis sa controlez ceea ce afisez pe LCD. LCD -ul poate afisa caracterele pe 2 randuri si 16 coloane, fiecare caracter fiind format din pixeli (o grila de 5x8 pixeli).Pinii LCD-ului:

1-GND

2-VCC (5V)

3-Vo (LCD contrast)=controleaza contrastul si stralucirea ,acestea sunt totodata reglate cu ajutorul potentiometrului

4-RS(Register Select) =diferentiaza comenzile de date(cand e LOW : trimitem comenzi pe LCD-setam cursorul intr-o anumita locatie,curatam display-ul,scrollam; cand e HIGH: trimitem date/caractere LCD-ului).

5-R/W(Read/Write) – 0 pentru output , 1 pentru input

6-EN (enable – cand e HIGH , lcd-ul proceseaza datele)

7-14 DataPins(D0-D7 Data Bus) = pinii care transporta datele trimise afisorului

15-A (anod)

16-C(catod)

Cu ajutorul functiei LiquidCrystal lcd(2, 3, 4, 5, 6, 7) se creaza un obiect LCD cu parametrii :

(RS, EN,D4, D5, D6, D7)

A screenshot of a computer

Description automatically generated with medium confidence

Am folosit 4 butoane pentru a directiona sarpele : spre stanga,dreapta,sus si jos.

A picture containing different

Description automatically generated

Initial, declar intr-o structura pct variabilele de tip intreg row,column, dir si un pointer de tipul structurii declarate ( pct \* next).

In variabilele declarate initial : ex : int butLeft =11; butonul care va directiona sarpele spre stanga este legat la placa la pinul 11, cu variabila int buttonStateL = 0; voi citi starea butonului; in functia setup initializez pinul butonului ca input cu ajutorul functiei pinMode(11, INPUT) .

Graphical user interface, text, application

Description automatically generated

Functia createSnake(int n) creeaza sarpele cu dimensiunea initiala n.

Initial , toata matricea jocului este libera.Aloc memorie cu ajutorul functiei malloc, care returnează un pointer valid catre blocul alocat pe heap sau pointer la NULL daca cererea nu poate fi îndeplinita.

tail = (point\*)malloc(sizeof(point)); Aloc dinamic memorie pentru sizeof(point) octeti, adresa returnata de malloc va fi un pointer catre o variabila de tip point.

Se initializeaza coordonatele cozii sarpelui : tail->row = 0; tail->column =60; tail->dir = left;

Setez pe true punctul pe care se afla coada in matrice mat[tail->row][tail->column] = true;

Pentru fiecare punct din care este format sarpele aloc memorie,daca dimensiunea acestuia e mai mare decat 1, se initializeaza head-ul ca primul pixel care nu e null din sarpe, altfel(daca sarpele are dimensiunea 1) head = tail.

Text

Description automatically generated

Functia punct() : declar un pointer de tipul structurii mentionate anterior point \*p; il initializez cu tail(adica coada sarpelui); declar o variabila boolean newPoint = true; si cat timp newPoint este true, se initializeaza linia si coloana pe care va aparea pixelul nou,pe care va trebui sa il manance sarpele,linia = un punct random din cei 16 si coloana =un punct random din cei 80.

Graphical user interface, text, application, email

Description automatically generated

Pe LCD se pot afisa 16 caractere pe 2 randuri, fiecare caracter fiind compus dintr-o matrice de 8 linii x 5 coloane de puncte/pixeli.

Functia drawMatrix() : cu ajutorul acesteia se afiseaza pe LCD jocul.

Afisarea fiecarui caracter este specificata de o matrice de 8 bytes, pentru fiecare linie si cei mai puțini semnificativi 5 biți din fiecare rând vor specifica care pixeli vor fi aprinși și care nu .

Daca pe o anumita linie si coloana din matrice este setat un pixel , formam caracterul de afisat si setez pe true booleanul black cu ajutorul caruia retin ca am de afisat un caracter pe LCD.

Aici se defineste care punct din cei 5 x 8 pixeli care trebuie luminat, aceasta se realizeaza scriind echivalentul binar al fiecarui rand si coloana.

Forma grafică efectivă a unui caracter afișat pe LCD este dată de conținutul memoriei CGROM (Character Generator Read Only Memory). Această memorie conține matricele de puncte pentru fiecare caracter (5x8 puncte sau 5x10 puncte – depinde de dispozitiv). Se va specifica pentru fiecare caracter o matrice de 8 octeți, câte unul pentru fiecare rând. Asadar, matricea jocului va avea dimensiunea de 16 linii : 2 randuri a cate 8 octeti si 80 coloane : 16 coloane a cate 5 octeti.

Cu functia lcd.createChar(count, character); creeez caracterul character la adresa count si il afisez.

Setez cursorul la coloana si randul la care trebuie afisat caracterul, cu functia lcd.write(byte(count)); scriu caracterul pe LCD , fac cast la byte.

A picture containing table

Description automatically generated

Functia growSnake() creste dimensiunea sarpelui,aloca memorie pentru fiecare nou punct din care e format acesta ,initializeaza tail-ul cu coordonatele noului punct : rand,coloana,directie si face totodata noul punct tail .

Text, letter

Description automatically generated

Functia moveHead() gestioneaza miscarea sarpelui dupa o directie. Daca directia este UP , atunci randul va scadea , pentru ca sarpele se va deplasa in sus in matricea LCD-ului, daca directia este la dreapta , coloana pe care se afla sarpele va creste ,analog pentru celelalte directii.

Text, letter

Description automatically generated

Text

Description automatically generated with medium confidence

Respectiv daca sarpele ajunge cu head-ul la coloana maxima a LCD-ului , adica >=80, se va initializa coloana head-ului cu 0. Daca sarpele ajunge cu head-ul la linia >=16, iarasi se va initializa linia head-ului cu 0.

Daca sarpele ajunge cu head-ul < coloana 0 , se va initializa coloana head-ului cu coloana maxima a LCD-ului , adica 80. Daca sarpele ajunge cu head-ul < linia 0 , se va initializa linia head-ului cu linia maxima a LCD-ului , adica 16.

Cat timp continua jocul, adica gameOver==false si tail!=head , daca sarpele ajunge sa isi atinga tail-ul, se seteaza gameOver la true, ceea ce inseamna ca s-a pierdut jocul.

Am pus regula ca in momentul in care jucatorul atinge scorul de 3 puncte, acesta sa castige jocul. Pe LCD se va afisa un mesaj si un caracter special,dat de functia like(). Respectiv se va afisa scorul obtinut de utilizator.

Diagram

Description automatically generated

Diagram

Description automatically generated

Daca sarpele colecteaza un nou pixel(head -ul se afla pe linia si coloana pe care se afla pixelul respectiv), scorul creste si se creeaza un nou punct.

In functia move() , declar un pointer de tipul structurii mentionate anterior point \*p; il initializez cu tail(adica coada sarpelui); matricea jocului boolean mat[16][80] este initializata cu false, adica toti pixelii din care e format LCD-ul sunt necolorati.

In functia move() ,sarpele se deplaseaza : initializez randul,coloana si directia sarpelui, , apelez functia moveHead().

A screenshot of a computer

Description automatically generated with medium confidence

Text, letter

Description automatically generated

Cu ajutorul functiei getButtonPress(int butLeft ,int butRight,int butUp ,int butDown ) verific care buton este apasat, citind starea acestora. Aceasta functie returneaza directia in care va merge sarpele.

Text

Description automatically generated

In functia setup() initializez pinii la care sunt legati butoanele ca fiind de intrare cu ajutorul functiei pinMode() . Setez LCD-ul cu numarul de 16 coloane si 2 randuri si apelez functia start() .

Functia setup() este folosita pentru a initializa variabile, pinii de intrare/iesire,folosirea librariilor,totodata ea este apelata o singura data la pornirea sau resetarea placii Arduino.

Dupa crearea functiei setup() , care initializeaza si seteaza valorile initiale ,functia loop() ruleaza intr-o bucla infinita, cum ii spune si numele, permitand programului sa se schimbe si sa raspunda la cererile utilizatorului.

In functia loop() verific care buton este apasat(cu ajutorul functiei getButtonPress(butLeft ,butRight,butUp , butDown) ) daca butonul apasat este in sus si daca directia sarpelui nu este in jos (capul ) ,sarpele isi poate modifica directia dupa cum ii spune butonul apasat. Daca butonul apasat este in dreapta si capul sarpelui nu este in stanga, adica sarpele nu se deplaseaza deja spre stanga,acesta isi poate modifica directia actuala.Analog pentru celelalte directii. Apoi se apeleaza functiile move() si drawMatrix() .

Text

Description automatically generated

Functia start() marcheaza inceputul jocului de snake.

Se seteaza gameOver la false, se elibereaza memoria alocata, se creeaza un sarpe de dimensiune initiala 6, se afiseaza o numaratoare inversa de la 5,apar mesajele de inceput si un caracter special smiley.

Graphical user interface, diagram

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Functiile like() si game\_Over() afiseaza un caracter special corespunzator finalului jocului, daca ai castigat – in cazul de fata am setat ca jocul sa se termine in momentul in care se aduna 3 puncte -respectiv daca ai pierdut -atunci cand sarpele isi mananca coada (tail=head). A picture containing text, circuit, electronics

Description automatically generated

A picture containing text, electronics

Description automatically generated

Cu ajutorul functiei free() eliberez memoria alocata . Funcția free primește ca și argument un pointer către blocul de memorie valid, care a fost anterior alocat .Funcția dezalocă zona respectivă de memorie, marcând-o ca fiind disponibilă pentru a putea fi ulterior realocată (refolosită) . După apelul funcției free programul nu trebuie să mai acceseze nici măcar un octet din blocul care a fost dezalocat sau să prespună ca acesta mai este încă valid.

Text

Description automatically generated

Link explicatie : https://youtu.be/jC2fq30JZpE