

FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

**Flappy Bird**

Student: VANCEA SORINA

Grupa: 30236

Profesor coordonator: Mircea Paul Muresan

Decembrie 01, 2020



FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

Cuprins

1. [REZUMAT 3](#_Toc60407913)
2. [INTRODUCERE 4](#_Toc60407914)
3. [PROIECTARE SI IMPLEMENTARE 7](#_Toc60407915)

[**3.1.** **Conectarea circuitului si a componentelor** 7](#_Toc60407916)

[**3.2.** **Implementarea propriu-zisa** 9](#_Toc60407917)

1. [REZULTATE EXPERIMENTALE 12](#_Toc60407918)
2. [CONCLUZII 15](#_Toc60407919)
3. [BIBLIOGRAFIE 16](#_Toc60407920)



FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

1. REZUMAT

Tema proiectului ales de mine este realizarea unui joc, numit Flappy Bird, insa intr-o versiune mai simplificata, cu componente minime deoarece imi doresc sa realizez acest joc pe un LCD 16X2, utilizand mediul de dezvoltare Arduino. Acest joc a fost testat si proiectat in simulatorul ThinkerCad, unde am realizat intregul circuit de care avem nevoie, cu urmatoarele componente: placuta Arduino Uno, LCD display 16x2, breadboard, potentiometru si un butoan.



FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

1. INTRODUCERE

* **Contextul temei si tendinte tehnologice**

Ideea propriu zisa a jocului, intitulat : **„Flappy Bird”** este controlarea unei pasari prin atingerea touchscreen-ului, ce face ca pasărea să zboare mai sus. Obiectivul jocului este să treci prin tuburi. Dacă lovești un tub, trebuie să o iei de la capăt, cu alte cuvinte, pasarea trebuie sa reuseasca sa zboare fara a lovi niste obstacole. Proiectul a fost realizat folosind mediul de dezvoltare Arduino, utilizand simulatorul ThinkerCad, versiunea online. Aceasta este ideea de la care am pornit, insa am pastrat doar ideea, adica jocul realizat de mine merge pe acelasi principiu, de a nu lovi obstacolul care se apropie, insa este realizat pe un LCD 16x2 care face ca posibilitatea folosirii unui touchscreen sa fie imposibila, „saritura” realizandu-se prin apasarea butonului, de asemenea si grafica relizata de mine este una simpla, reinterpretand celebra pasare din Flappy Bird.

* **Descrierea pe scurt a solutiei propuse**

Pentru realizarea acestui proiect este necesara proiectarea unui program prin care placuta Arduino controleaza cu ajutorul unui buton miscarile de SUS sau JOS pe care „pasarea” le realizeaza pe ecranul LCD-ului. Pentru acest proiect, placuta va trebui sa citeasca starea butonului, astfel aceasta va trebui sa stie cand butonul este apasat de catre utilizatorul jocului. Acest lucru se realizeaza cu ajutorul functiei digitalRead(), care va citi starea butonului. Am ales sa replichez pasarea cu un caracter care sa semene cu un extraterestru, si am reprezentat obstacolele prin caracterul: `-`, de fapt ceea ce am facut a fost sa declar 2 siruri diferite cu astfel de caractere , unul pentru randul de sus si unul pentru randul de jos. Scopul acestui joc este ca „pasarea” sa reuseasca sa treaba prin aceste obstacole fara a le atinge, in acest fel castigandu-se jocul.



FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

* **Definirea problemei de rezolvat**

Problema de rezolvat consta in transmiterea unor string-uri care sa ilustreze traseul pe care „pasarea” noastra va trebui sa il parcurga, obstacolele de care va trebui sa se fereasca:



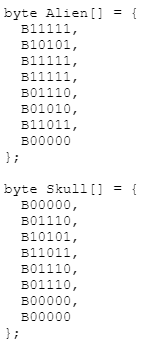


, dar si al câtelea caracter din stringul de mai sus este fiecare obstacol in parte(reprezentat de o linie orizontala), lucru declarat prin sirurile de int-uri de mai jos:

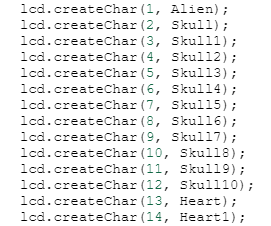




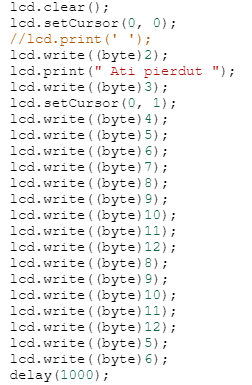
De asemenea, caracterele care apar pe parcursul jocului, adica caracterul care tine locul pasarii sau caracterele care apar atunci cand jucatorul a castigat(caracterele sub forma de inima) sau a pierdut jocul (caracterele sub forma unui craniu) sunt declarate cu ajutorul Custom Character:



, dupa care sunt create in functia **setup():**



, iar mai apoi in functie de locul in care vrem sa le afisam sunt scrise cu ajutorul functiei **write(),** spre exemplu pentru a afisa pe ecran mai multe caractere sub forma unui craniu in momentul in care utilizatorul a pierdut jocul, vom scrie in functia finalGame():



, unde pe langa mesajul cu „Ati pierdut” pe care utlizatorul jocului o sa il primeasca i se vor afisa si mai multe caractere sub forma unui craniu, fiind sugestiv pentru rezultatul obtinut de acesta. Analog, pentru castigarea jocului se va scrie in functia castigGame() pe langa mesajul pe care dorim sa il printam pe ecran prin care anuntam utilizatorul ca a castigat jocul si mai multe caractere sub forma unor inimi.



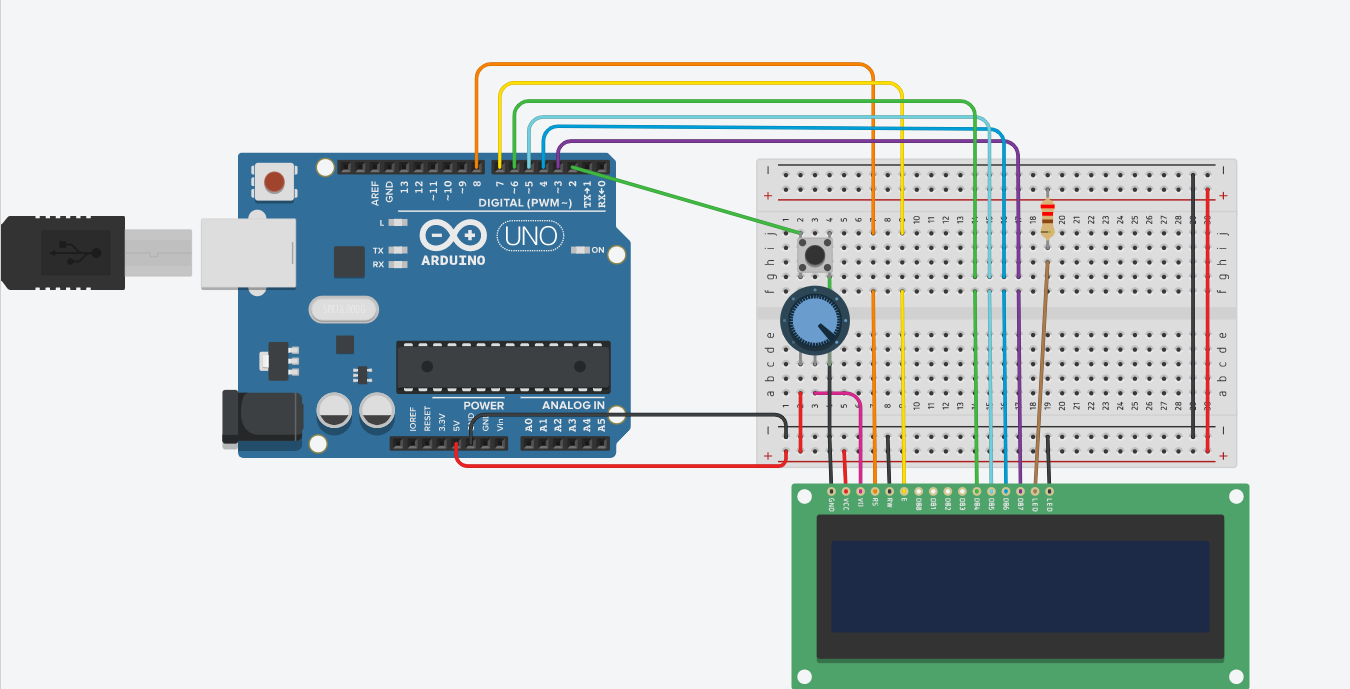
FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

1. PROIECTARE SI IMPLEMENTARE

## **Conectarea circuitului si a componentelor**

Mai jos putem observa schema generala a proiectului, realizata in simulatorul ThinkerCad:



Componentele de care avem nevoie pentru ca proiectul sa functioneze sunt descrise mai jos:

* **Placuta Arduino UNO**
* **LCD display 16x2**
* **Breadboard**
* **Potentiometru**
* **Buton**
* **Fire**
* **Sursa de alimentare de 5V**



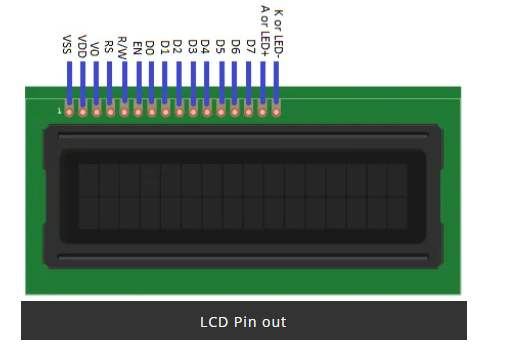
FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

In ceea ce priveste structura circuitului, pentru a controla ceea ce voi afisa pe LCD, am folosit libraria **<LiquidCrystal.h>,** unde avem de afisat 2 randuri si 16 coloane:



, fiecare caracter fiind format din pixeli(5X8 pixeli). LCD-ul cu display 16X2 are urmatorii pini:



, unde daca dorim conectarea la placuta Arduino vom avea:

* VSS -> GND
* VDD -> 5V
* V0 -> mijlocul potentiometrului de 10k
* Rs -> pin 7
* R/W - > GND
* E -> pin 6



FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

* D4 -> pin 5
* D5 -> pin 4
* D6 -> pin 3
* D7 -> pin 2

Astfel, avand in vedere cele prezentate mai sus, am conectat pinii RS si E la pinii 8 si 7 de la Arduino Uno, pinii de date D4, D5, D6, D7 la pinii 6, 5, 4, 3, iar butonul a fost conectat intre GND si pinul 2. De asemenea, am conectat si un potentiometru pentru a regla contrastul LCD-ului.

## 

## **Implementarea propriu-zisa**

In realizarea acestui proiect am folosit intreruperile, pentru a putea opri jocul in momentul in care o actiune noua s-a produs, fie ca utilizatorul jocului a castigat, fie a pierdut. Astfel, avem nevoie ca in momentul in care jocul se opreste(indiferent de cauza care a produs acest lucru), microcontrollerul sa opreasca programul pe care il executa si sa inceapa sa se ocupe doar de intreruperea noastra, facand acest lucru prin activarea bitului corespunzator acelei intreruperi. Pentru a utiliza acest lucru, folosim functia **attachInterrupt(),** functie care are rolul de a atasa o functie ISR la o intrerupere externa, inlocuind orice functie precedenta care a fost atasata anterior si de a activa intreruperea. Acest lucru il relizam in functia **setup():**



, unde:

* **digitalPinToInterrupt(2)** – este numarul intreruperii atasata pinului digital
* **btn\_ISR** – este procedura de tratare a intreruperii, o functie declarata la finalul programului, care verifica daca butonul este apasat si care este starea anterioara a butonului, astfel dorim sa stim daca butonul a fost apasat acum, sau a ramas de la vechea apasare pe buton.
* **CHANGE** – este conditia de declansare a intreruperii, astfel, astfel la orice schimbare de nivel pe pin se va declansa o intrerupere.

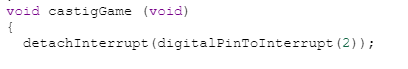


FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

De asemenea, cu ajutorul functiei **detachInterrupt()** vom dezactiva intreruperea cu numarul specificat ca parametru, utilizata atat in functia finalGame(), cat si in castigGame():



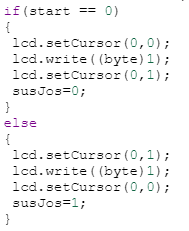


In ceea ce priveste functia btn\_ISR() pe care am prezentat-o mai sus, de fiecare data cand se intra in ea, adica are loc o intrerupere, verificam starea butonului fata de ultima apasare, astfel vedem daca s-a apasat butonul sau nu:



, daca acest buton a fost apasat, atunci vom modifica stareaAnterioara cu LOW , modificam semnalul pinului care contine ledul , in functie de start(daca start = `0` inseamna ca acum am inceput jocul):

si verificam daca start == 0 , atunci „pasarea” noastra este pozitionata initial (la intrarea in functie) pe linia de sus a LCD-ului si dupa iesirea din if ii vom schimba pozitia pe linia de jos a LCD-ului; in cazul in care start != 0, atunci „pasarea” initial este pe linia de jos si dupa iesirea din else ii vom schimba pozitia:

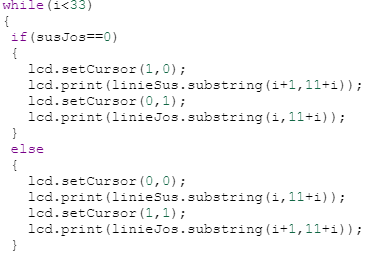




FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

In ceea ce priveste loop() -ul functiei, aici are loc parcurgerea propriu-zisa prin „obstacolele” pe care eu le-am declarat sub forma unor stringuri de `-`, astfel caracterul cel mai indepartat din stringuri este pe pozitia 32, deci vom urmari ca atata timp cat nu am ajuns la ultimul caracater atat de pe linia de sus, cat si de pe linia de jos(adica inca avem obstacole), sa verificam daca „pasarea” este pozitionata pe linia de sus sau pe linia de jos a LCD-ului, daca este pe linia de jos, atunci vom seta cursorul in stanga jos si vom printa urmatoarele „obstacole”, apoi vom muta cursorul pe linia de sus, o pozitie mai la dreapta, iar daca pasarea este pozitionata pe linia se sus, atunci setam si cursorul la (0,0), printam urmatoarele caracatere si modificam din nou pozitia cursorului pe linia de jos. Acest lucru a fost scris in cod astfel:



De asemenea, in loop() facem verificarile propriu-zise pentru a stii cand jocul este castigat si cand este pierdut, astfel vom parcurge sirul cu cel mai mare numar de „obstacole” , adica de `-` declarat si daca ne lovim de unul dintre obstacole, atunci jocul a ajuns la final, iar daca ajungem la ultimul obstacol si nu ne-am atins de el, atunci am castigat jocul

Pentru a controla viteza cu care „pasarea” se deplaseaza am declarat o variabila numita viteza pe care am initializat-o cu 200 si la care ii fac un delay.

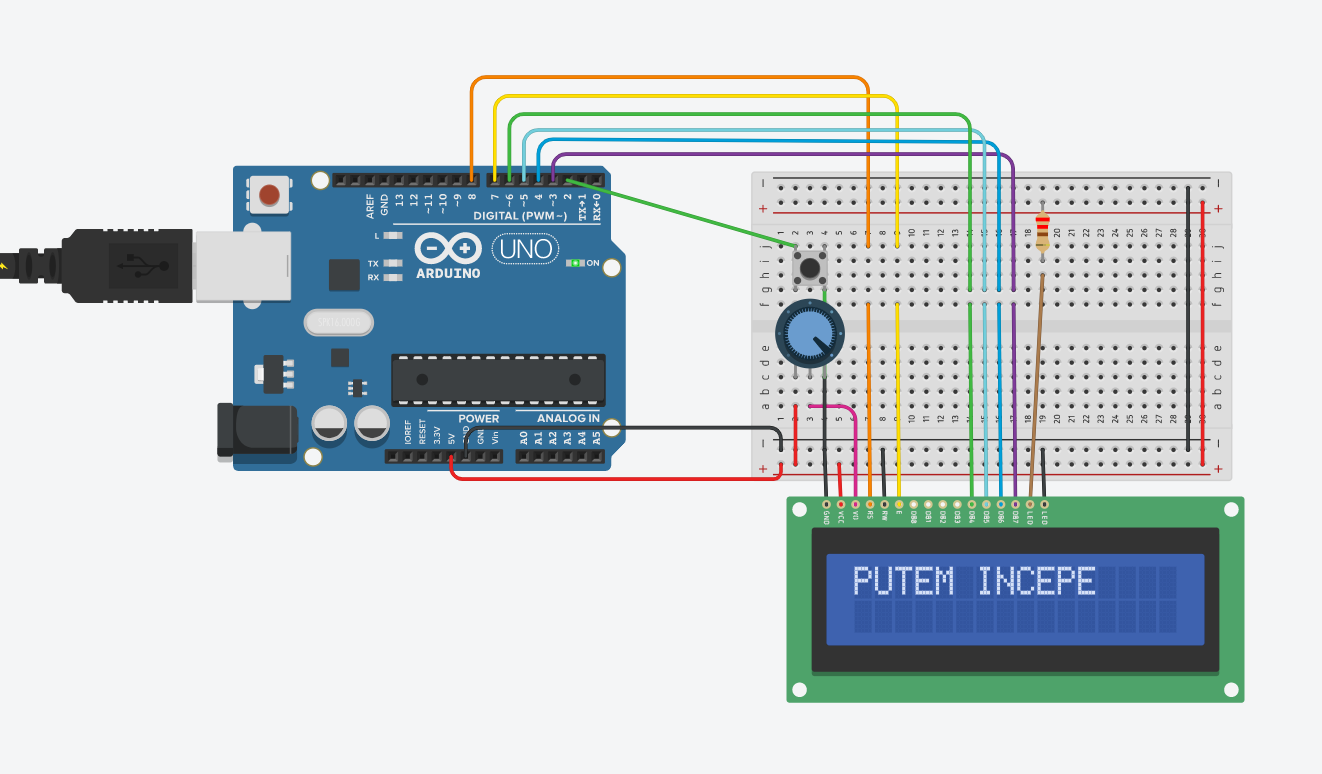


FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

1. REZULTATE EXPERIMENTALE

Dupa ce am relizat toate conexiunile prezentate mai sus si adaugam codul sursa in simulator, odata ce apasam butonul „**START** **SIMULARE**” primul lucru care va aparea pe LCD este un mesaj prin care sa stim ca aplicatia este pregatita sa inceapa („PUTEM INCEPE”):

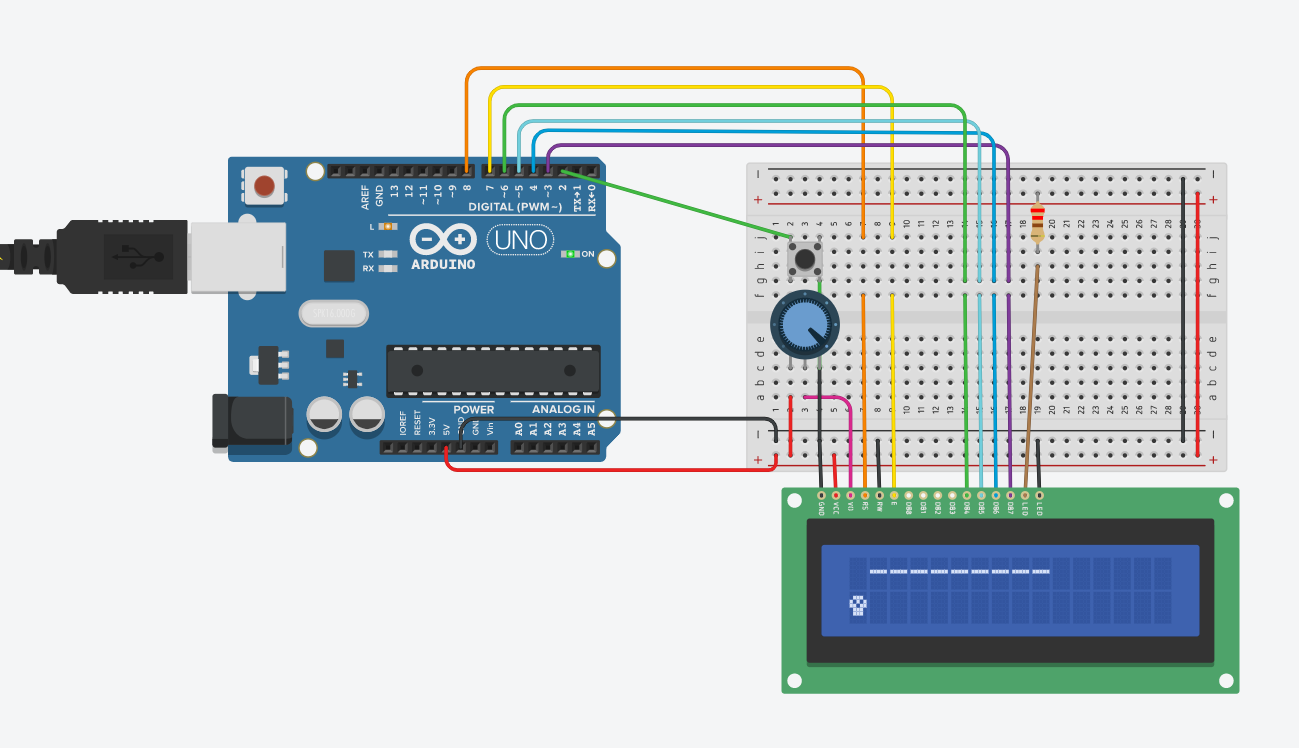


Odata ce acest mesaj a fost afisat, pozitia „pasarii” poate fi schimbata folosind pin-ul de intrerupere, cu ajutorul apasarii butonului , in cazul in care pasarea se afla pe linia de sus a LCD-ului aceasta poate fi mutata in partea de jos apasand butonul si vice-versa.

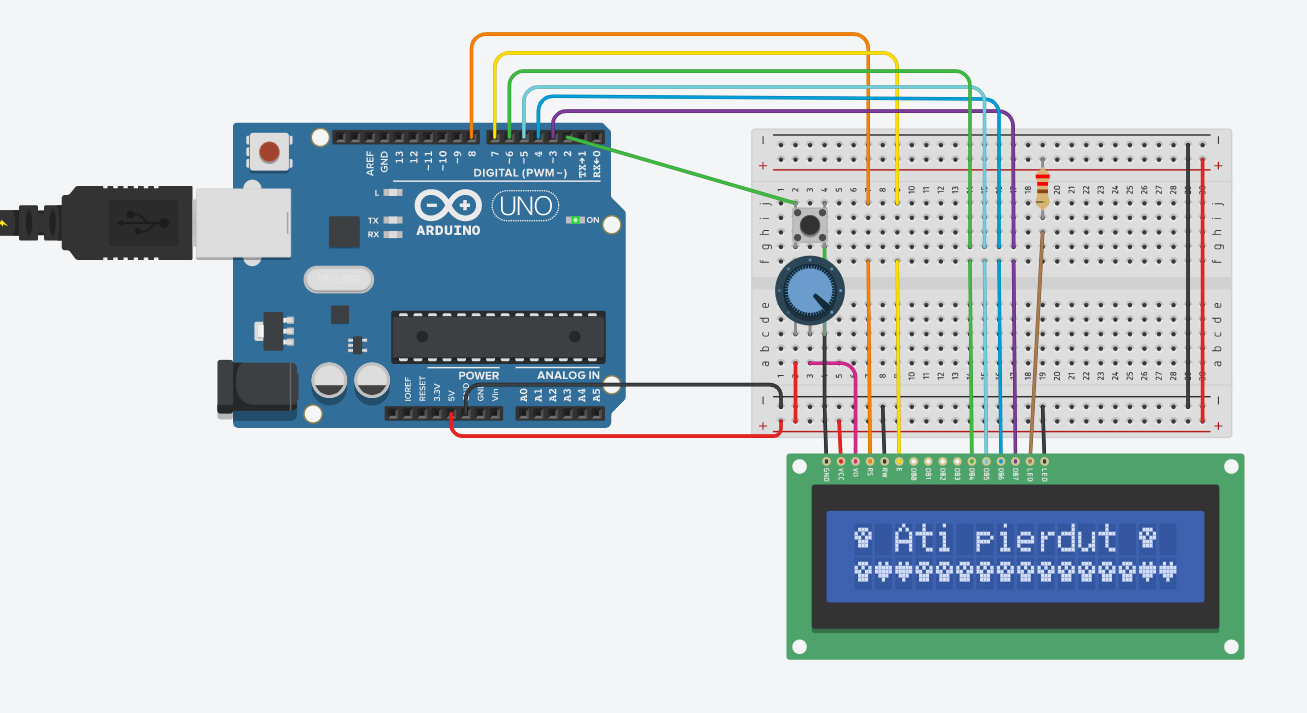


FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE



In cazul in care „pasarea” va atinge oricare dintre obstacole, utilizatorul jocului va pierde acea runda, si i se va afisa urmatorul mesaj:

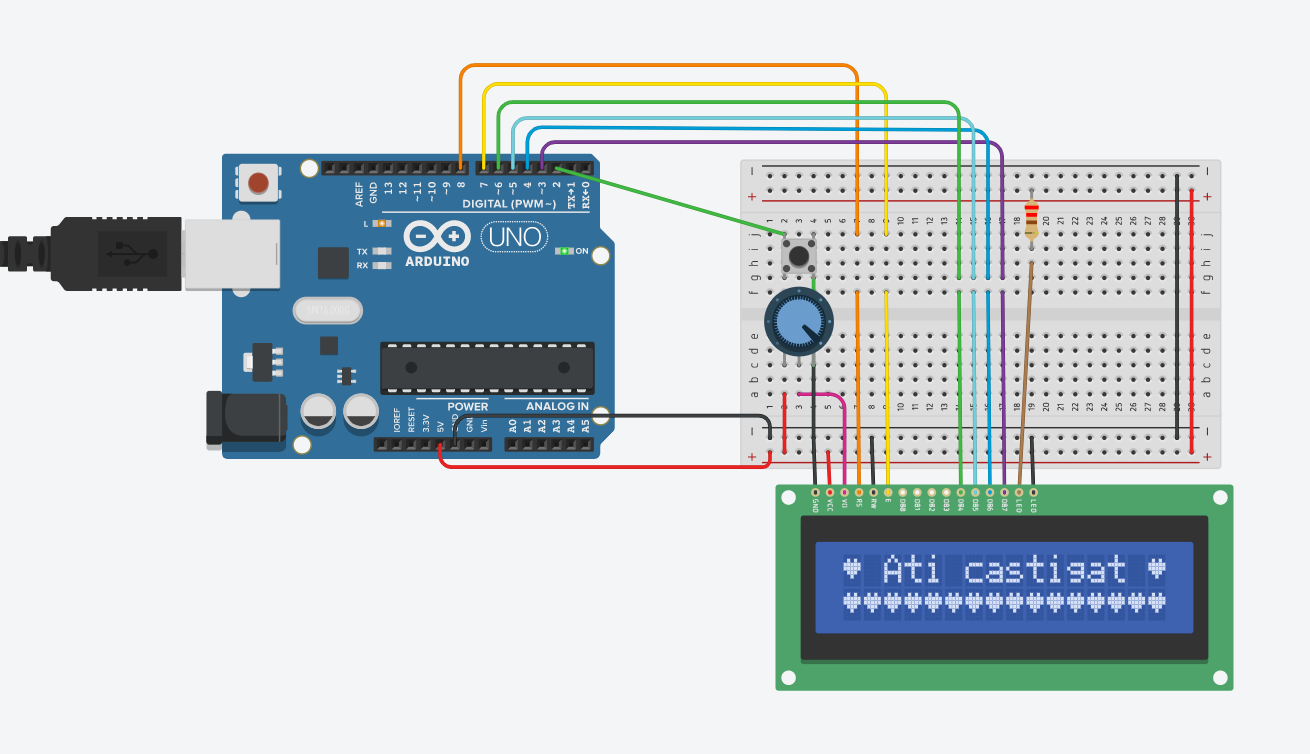




FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

Deoarece aceasta este o versiune mai simplificata a jocului original, in care nu exista atat de multe obstacole, atunci cand jucatorul trece de ultimele obstacole fara sa le loveasca acesta va castiga jocul si i se va afisa urmatorul mesaj:





FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

1. CONCLUZII

Proiectul pe care eu l-am prezentat este o versiune simplificata a celebrului joc, cunoscut sub denumirea de „Flappy Bird”, in care am folosit placuta Arduino UNO, un LCD cu display 16X2 si un buton de apasare. Acest joc nu reproduce jocul original, insa incerca sa ofere o idee mai simpla in care acest joc ar putea fi realizat si bine-inteles intr-un buget mult mai mic.

In ceea ce priveste o dezvoltare ulterioara a proiectului, ar fi incercarea reproducerii cat mai apropiate a jocului original, folosind de aceasta data un LCD cu touchscreen, care sa permita controlul pasarii prin atingerea cu degetul si eventual sa monitorizeze scorul pe care jucatorul a reusit sa il obtina, actualizand mereu best scorul, astfel jucatorul sa fie motivat pentru a-si bate propriul record.

In concluzie, consider ca aceast joc este unul care poate creea foarte usor dependenta, fix asta fiind unul dintre principalele dezavantaje ale jocului de care creatorul jocului original s-a batut, insa popularitatea lui a dus la mai multe jocuri replica si versiuni clonate ale acestuia, jocul in sine fiind un adevarat succes.



FACULTATEA DE AUTOMATICA SI CALCULATOARE

CATEDRA: CALCULATOARE

1. BIBLIOGRAFIE

[1] [https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/336-3.pdf](https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/336-3.pdf%20)

[2] <https://www.youtube.com/watch?v=bjblQqfjdWM>

[3[] https://stackoverflow.com/questions/42781037/meaning-of-setcursorx-y-coordinates](%5d%20https:/stackoverflow.com/questions/42781037/meaning-of-setcursorx-y-coordinates)

[4] <https://omerk.github.io/lcdchargen/>

[5] [Arduino LCD 16X2 Interfacing | Tutorial (electronicshobbyists.com)](https://electronicshobbyists.com/arduino-lcd-interfacing-tutorial/#:~:text=LCD%20is%20the%20most%20commonly%20used%20part%20when,that%20LCD%20has%2016%20columns%20and%202%20rows.)