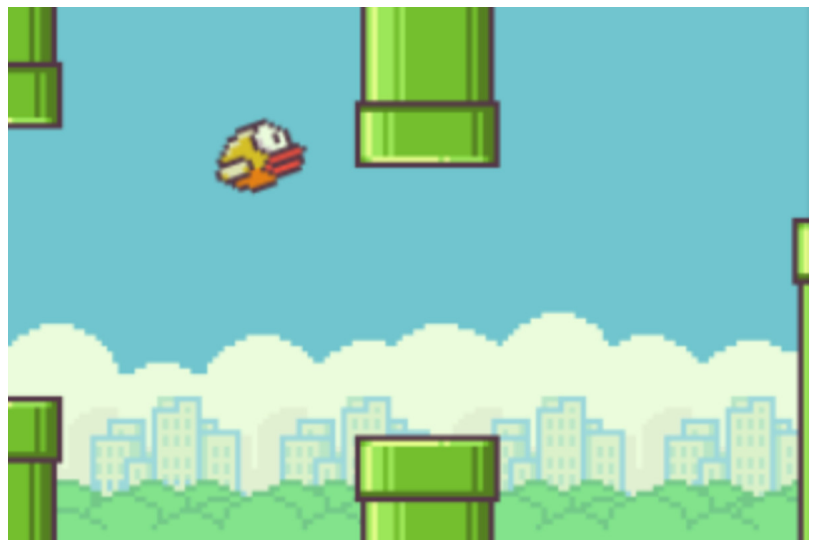


# Ioan MĂRĂCINEANU (78401) - Flappy Bird

Autorul poate fi contactat la adresa: **Login pentru adresa**

## Introducere



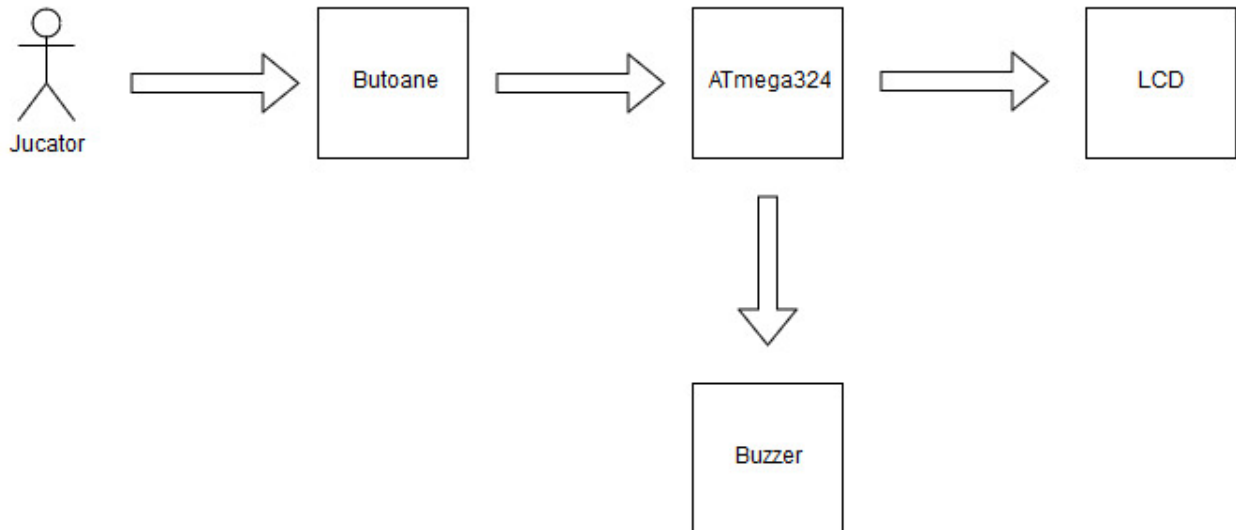
Ideea mea a fost sa iau un joc simplist, caruia sa ii implementez o varianta mai complexa. De aceea am ales Flappy Bird, un joc pe mobil, care a fost viral in 2013.

In acest joc, controlezi o pasare prin atingerea ecranului, ce face ca pasarea sa zboare mai sus, obiectivul fiind depasirea unor obstacole sub forma de tuburi (pereti printre care trebuie sa zboare). Daca lovesti un obstacol o iei de la capat.

Am adaugat functionalitati in plus, cum ar fi posibilitatea distrugerii obstacolelor, prin aruncarea unor proiectile explozibile (oua), precum si o abilitate care ii permite pasarii sa treaca prin pereti (Ghost Mode). Desigur, daca distrugi o parte dintr-un obstacol, atunci nu vei mai primi puncte pentru depasirea lui. De asemenea, abilitatea are un timp de incarcare (cooldown).

Scopul proiectului este obtinerea unui joc amuzant si incitant, in special pentru cei mai competitivi, intrucat desi jocul pare destul de simplu, obtinerea un scor bun este dificila.

## Descriere generală



Jucatorul va folosi 3 butoane: cate unul pentru deplasarea pasarii, tragerea in obstacole si folosirea abilitatii. Scena 2D va putea fi vizualizata pe un ecran LCD. De asemenea, voi folosi un buzzer pentru redarea sunetelor.

## Hardware Design

### Lista de piese:

- Placa si componentele din pachetul de baza
- Butoane
- Fire de conexiune
- translator 5V-3.3V (pentru LCD)
- regulator LM1086
- LCD SPI 2.8 240×320 RGB
- Buzzer

### Schema electrica:



## Software Design

Am folosit win-avr ca mediu de dezvoltare si HIDBootFlash pentru scrierea pe microcontroller.

Am utilizat cateva biblioteci din laborator: `avr/io.h`, `util/delay.h`, `avr/interrupt.h`; de asemenea, bibliotecile pentru LCD: `ili9341.h` si `ili9341gfx.h`. [1]

### Detalii generale de implementare:

- In primul rand, am definit cateva structuri care contin parametrii pentru obiectele continute in joc: pasarea, zidurile (fiecare zid are o deschidere la mijloc, prin care trebuie sa treaca pasarea) si proiectilul (devine activ la apasarea butonului de tragere). Am o functie care initializeaza majoritatea parametrilor.
- La inceputul jocului se afiseaza cateva mesaje introductive (numele, instructiunile de control).
- Miscarea pasarii se face doar pe verticala (ca in jocul original), deci am variabile ce definesc componentele verticale pentru viteza si acceleratie (acceleratia e constanta - in jos).
- Pentru tratarea coliziunilor dintre ziduri si pasare (sau ziduri-proiectil), am o functie care parcurge lista de tuburi si verifica daca are loc vreo coliziune. In caz ca proiectilul loveste un zid, acesta este sters. Am considerat toate obiectele dreptunghiuri, pentru simplificare.

### Interactiunea cu hardware-ul:

- Am configurat intreruperi pentru tratarea apasarii celor 3 butoane: de exemplu, butonul 1 doar seteaza viteza pasarii la o valoare pozitiva constanta, practic ii da un impuls, oferind sensatia ca aceasta zboara in sus.
- Intrucat abilitatea speciala are un cooldown si o anumita durata cat timp poate fi activa, am avut nevoie sa configurez un timer, asa ca am folosit Timerul 1 in modul CTC, pentru a numara unitatile de timp scurse. Am configurat LED-ul de pe placa de baza sa se aprinda atunci cand s-a incarcat abilitatea, adica poate fi activata cu butonul 3.
- Jocul este dinamic, astfel ca pozitia obiectelor pe ecran trebuie modificata constant. In principiu, in bucla while principala afisez obiectele pe pozitia lor curenta, astept un anumit timp (delay), dupa care sterg obiectele (adica le redesenez in culoarea fundalului) si trec la urmatorul cadru.
- Intrucat LCD-ul este destul de lent, miscarea tuburilor trebuie executata eficient: la fiecare mutare a unui tub (cu o unitate la stanga), adaug o linie la stanga tubului, iar linia sa cea mai din dreapta o sterg.
- Fiecare actiune importanta genereaza un mic zgomot redat de Buzzer, prin functia PLAYNOTE caruia ii transmit durata si frecventa sunetului, metoda asemanatoare cu cea din laborator [2].

## Rezultate Obținute

It works!! <https://www.youtube.com/watch?v=Nxvutpsp93Y>



## Concluzii

Consider ca proiectul a fost interesant si util pentru intelegerea si aprofundarea cunostintelor dobandite in cadrul laboratorului, precum si pentru familiarizarea cu lucrul manual. Per total o experienta placuta.

## Download

[ioan\\_mrc\\_arhiva\\_pm.zip](#)

## Bibliografie/Resurse

### Resurse Software

[1] LCD libs:

<https://community.atmel.com/projects/ili9341-library-drive-22-tft-displayderived-adafruit-tft-library-ili9340-type-controller>

[2] PLAYNOTE: <https://www.petervis.com/C/pizo%20speaker/pizo%20speaker.html>

[3] Laboratoarele de PM, in principal acesta: <http://cs.curs.pub.ro/wiki/pm/lab/lab2>

### Resurse Hardware

<http://optimusdigital.ro/>

- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2018/rbarbascu/flappy>



Last update: **2021/04/14 15:07**