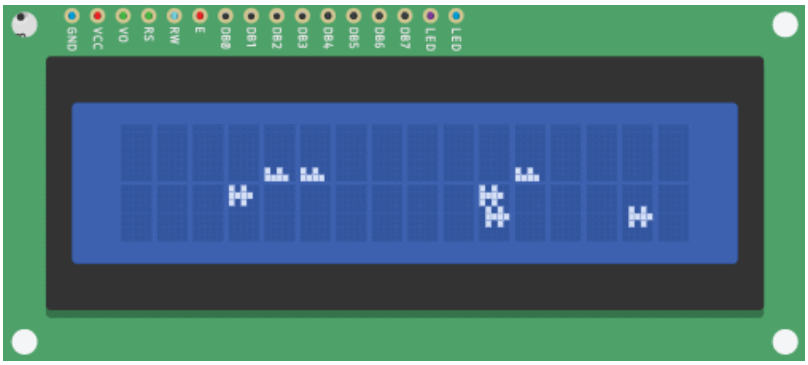
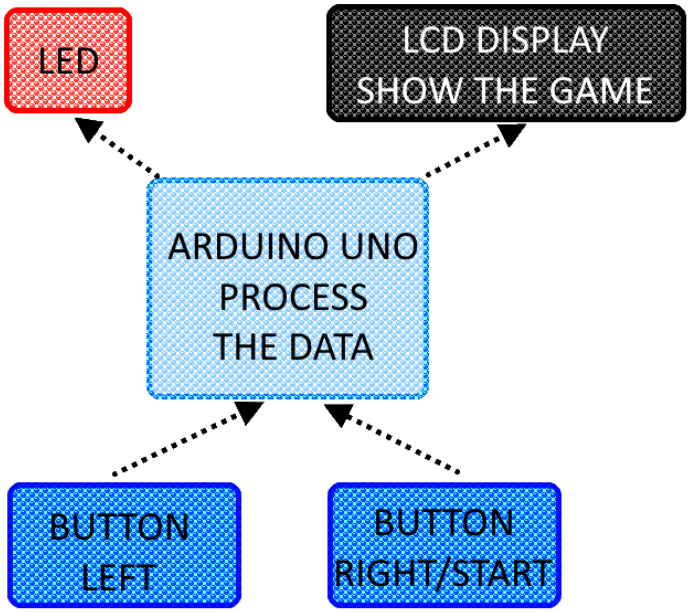
Carmageddon: The LCD Game

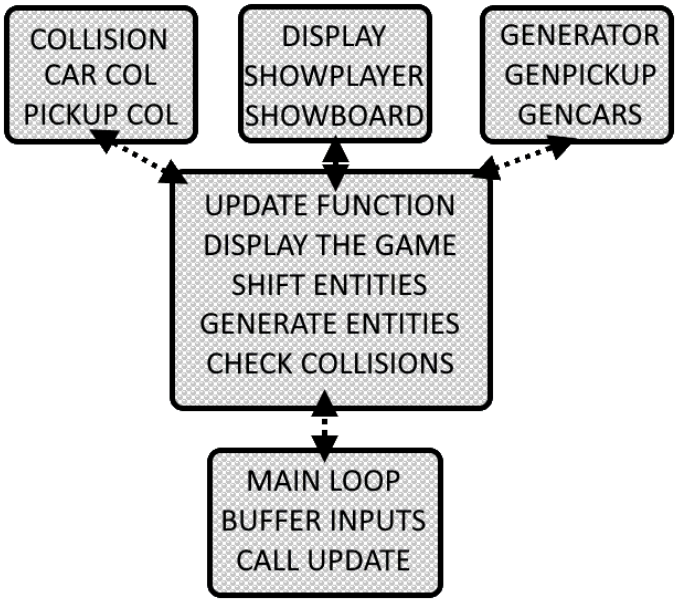
Introducere



Proiectul meu este un joc realizat pe lcd utlizand o placa arduino uno. Jocul consta in navigarea cu o masinuta pe un lcd si distrugerea inamicilor prin ciocnire pentru a obtine un scor mai mare insa pentru fiecare ciocnire vine un cost de reparatii. Inspiratia mi-a venit de la nostalgia fata de jocurile de pe platformele limitate cum ar fi nokia si gameboy.

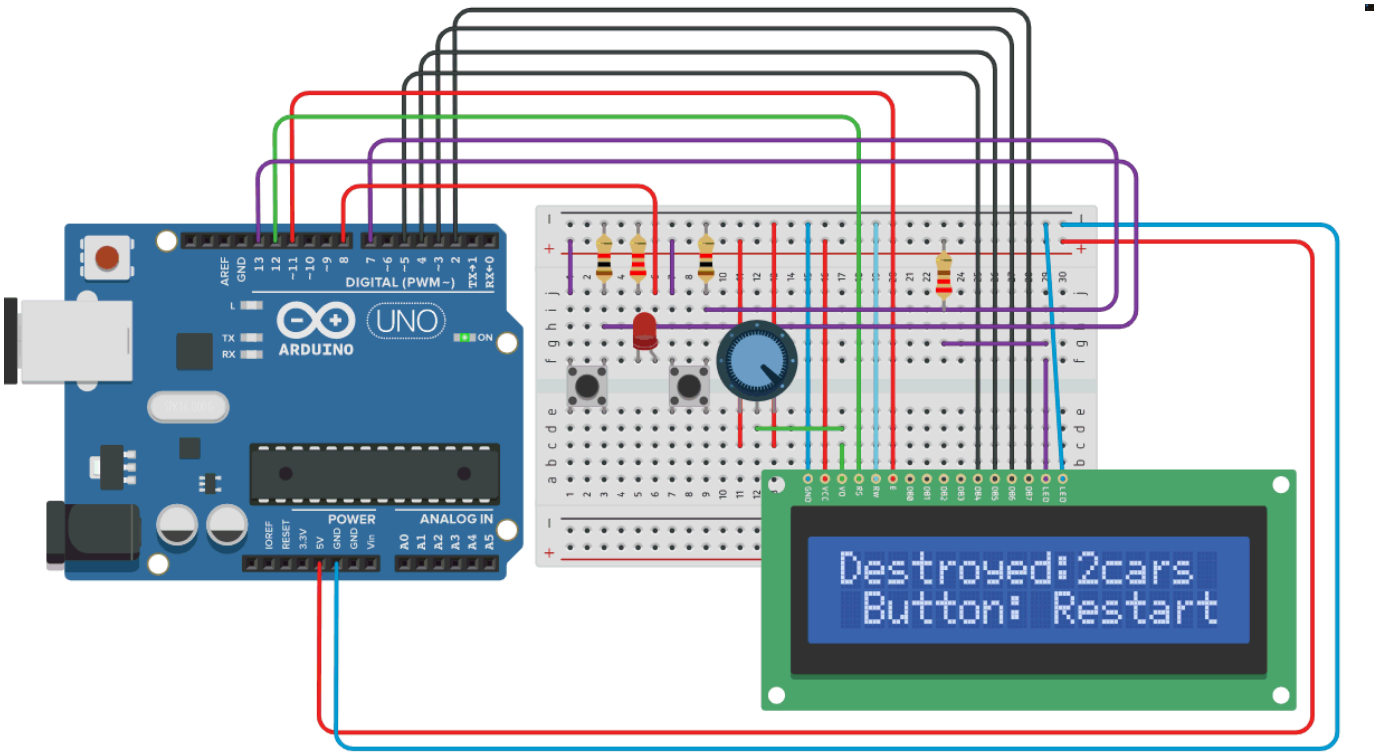
Descriere generală





Jucatorul va comunica cu jocul prin intermediul a doua butoane. Fluxul de date este procesat de arduino unde in dependenta de starea jocului va genera un output corespunzator.
La nivelul software avem doua loop-uri principale. Loop-ul main este dirijat de frecventa ceasului de pe placuta iar loop-ul secundar este rulat la fiecare N (250) milisecunde. In loop-ul principal are loc buferizarea input-urilor si aplearea loop-ului secundar la un interval stabilit.

Hardware Design

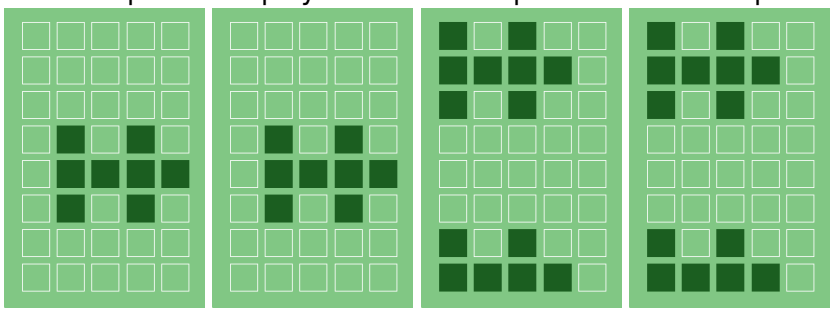


Lista de piese:

- Arduino UNO
- Display LCD 2×16
- 2 Butoane
- Potentiometru
- 1 LED

Software Design

Pentru implementarea temei am utilizat libraria “**LiquidCrystal**” care mi-a oferit un mod mai intuitiv de interfatare cu un display lcd2x16. Intial in program are loc setarea bitilor pentru fiecare caracter aditional pentru display. Acestea fiind pozitiile masinilor pe 3 linii si combinatiile intre ele.



Din motivul prezentei limitei de 8 caractere aditionale pe un cip de lcd am recurs la metode diferite de a adauga peste limita. Una a fost incarcarea unor diferite charset-uri alta a fost utilizarea alfabetului deja prezent pe cip. Am folosit a doua metoda, astfel am utlizat '*' pentru a reprezenta o explozie sau '$' pentru a reprezenta starea financiara a jucatorului.

In prima etapa a rularii programului fac o buferizare a inputurilor de la butoane in acest mod garantez faptul ca orice input primit de la utilizator va ajunge in functia de update a jocului. Buferizarea are loc in loop-ul principal print intermediul a cateva variabile globale: button, button2. Butoanele pot avea 3 stari + doua intermediare, starile : pressed (**BTN_HLDNG**), hodling (**BTN_PRSSD**), released (**BNT_RLSED**)

In loop-ul de update primim aceste inputuri ca pe baza lor sa facem diferite procesari. Datorita starilor intermediare putem detecta comportamentul exact cum ar fi cand a fost apasat butonul pentru a realiza un one-time-event sau putem verifica daca butonul este tinut pentru a face ceva continuu cum ar fi schimbarea de lane-uri din joc sau chiar sa detectam un double-press. La tinerea butonului din stanga masina jucatorului se deplaseaza in stanga, la apasarea celui drept-dreapta.

Toata tabla de joc este o matrice 2×16 unde fiecare element de pe placa este reprezentat printr-un index corespunzator. In cazul masinilor acest index se calculeaza prin operatia ori binar.

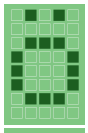
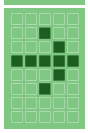
Pozitia jucatorului se poate schimba la fiecare tick, insa masinile de pe tabla se deplaseaza ba la fiecare al 3-lea sau al 4-lea tick. Motivul acestei diferente este de a crea timp pentru a face ocolirea masinilor mai usoara. Dupa schimbarea pozitiei are loc afisarea tablei de joc folosind functiile write din biblioteca “**LiquidCrystal**”. Pentru fiecare byte din tabela de joc ii corspunde un caracter.

La aceasta etapa restul operatiilor se realizeaza la interval mai redus (al 3-lea 4-lea tick din update).

Functia de siftare muta fiecare element de pe tabela de joc cu o pozitie. Deoreace la mutarea tabeleui de joc vor aparea goluri generam cu autorul functiilor random elemente noi cum ar fi masini si

pickup-uri.

Dupa ce s-au generat toatle elementele - apelam functia de check pentru colisiuni a jucotorului cu lumea. In fucnctia **carcrash()** se fac mai multe check-uri. Avem check pentru pickup-uri; Fiecare pick-up are puterile sale:

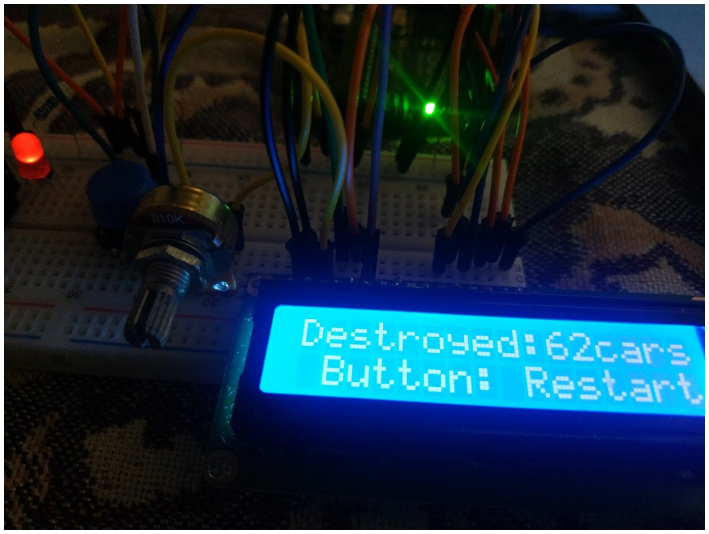
-  Distruge toate masinile din radar si ofera bani jucatorului.
-  Mareste viteza jocului si face masina invincibila pentru o durata de 150 de tick-uri.

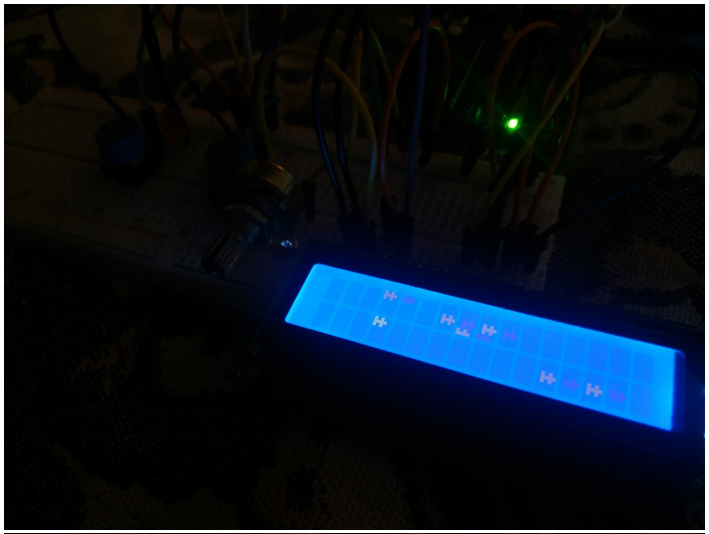
Si se face check pentru masini:

- Daca jucatorul nu se ciocneste cu o masina de pe linia sa
- Ciocnire → marim scorul → afisam scorul → verificam daca platim sau suntem platiti → afisam contul.

Scopul jocului este sa distrugi cu atat mai multe masini este posibil fara ca sa ramai fara bani pentru reparatii. In cazul in care ramanem fara bani afisam mesajul de game over **death_message();** si cream un prompt de restart.

Rezultate Obţinute





Demo: Carmageddon

Concluzii

In realizarea acestui proiect m-am convins cat de importanti sunt rezistorii .

Download

Carmageddon: The Racing Game Free Download 2021

Jurnal

Bibliografie/Resurse

Arduino KIT (Do Not Recommend)
LCD char gen
Tinker my circuits

Export to PDF