

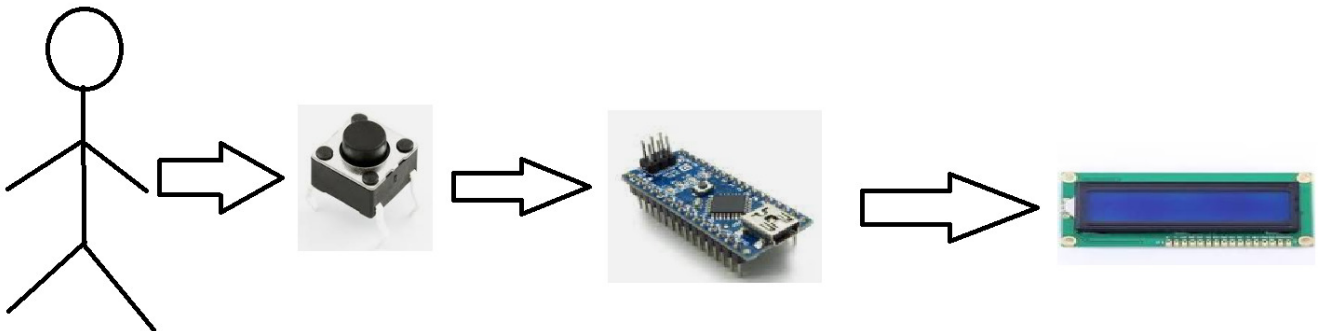
# Dodge The Wall

Realizat de Floca Andrian-Alexandru-Irinel

## Introducere

Proiectul acesta reprezinta un joc in care trebuie sa facem un caracter sa se fereasca de anumiti pereti. Cu ajutorul unui LCD 16x2 afisam acest joc, iar cu ajutorul unui buton putem sa controlam caracterul din joc.

## Descriere generală



Butonul interactioneaza cu Arduino-ul prin intermediul pinului 9. Initial se leaga LCD la Arduino prin intermediul pinilor descriși la Hardware Design. Cu ajutorul librăriei LiquidCrystal.h initializam o variabila care reprezinta LCD-ul nostru. Prin diferite apeluri de functii afisam ce dorim pe ecran.

## Hardware Design

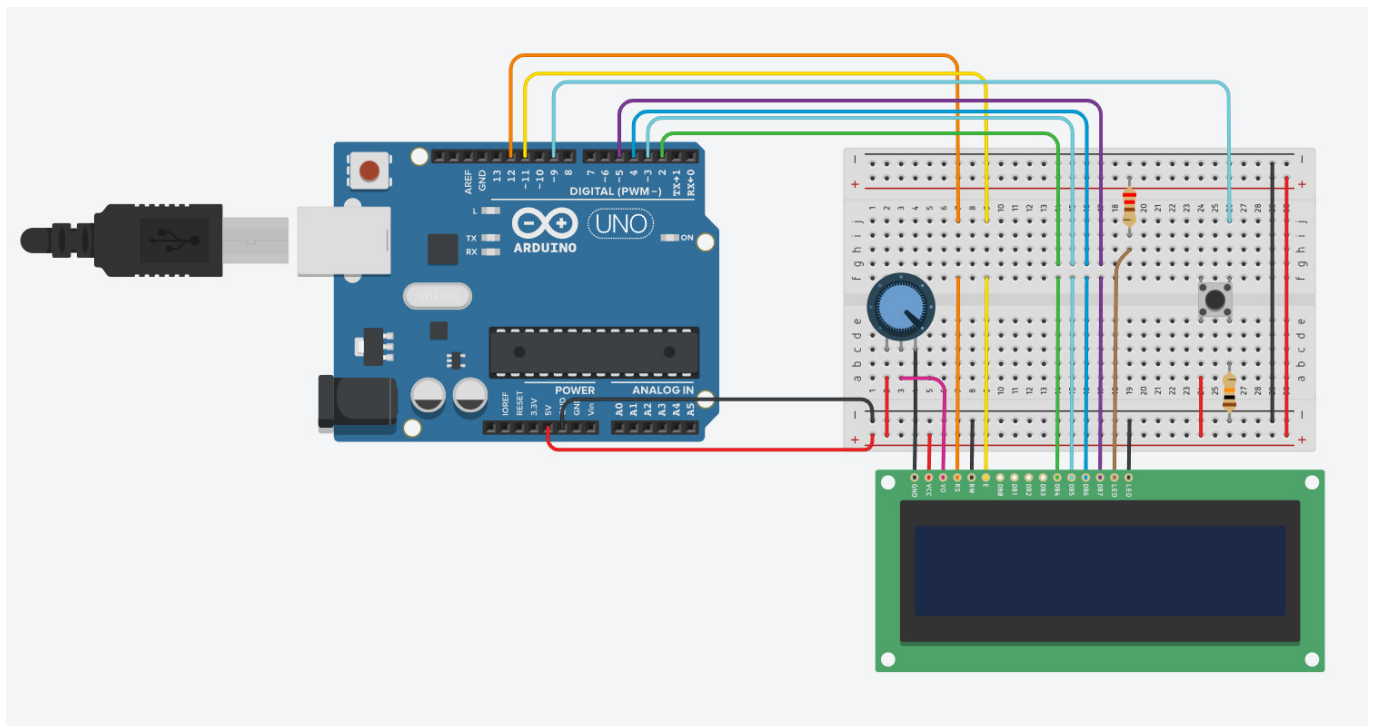
- Lista componente

1. Arduino Uno
2. Ecran LCD 16x2
3. Potentiometru
4. Buton
5. LED Rosu, LED Verde
6. Buzzer
7. Senzor Distanta
8. Diferite rezistente

## 9. Materiale constructii(placa PCB, fire)

- Conexiune pini

Pini componente	Pini Arduino
RS	12
EN	11
D4	2
D5	3
D6	4
D7	5
Button	9



In aceasta reprezentare hardware a proiectului am folosit un Arduino Uno deoarece aplicatia pe care am folosit-o(TinkerCAD) nu avea Arduino Nano. Pini din reprezentare sunt legati ca pe proiect.

## Software Design

-Librarii folosite:

Am folosit libraria LiquidCrystal.h deoarece aceasta are multe functii cu ajutorul careia putem sa controlam afisajul LCD-ului.

-Explicatii cod:

Am definit un LCD in care argumentele sunt pini descrisi la definirea pinilor.

In functie setup am initializat un LCD cu dimensiunile acestuia(16 , 2).

Cu ajutorul functiei createChar am definit 2 caractere speciale(unul pentru jucator si unul pentru perete).

Am initializat pinul butonul(acesta fiind de input).

Cu ajutorul variabilei globale sceneManager tinem cont de scena pe care vrem sa o prezentam:

0 → startScene(scena in care este prezentat titlul jocului).

1 → mainMenu(scena in care se prezinta instructiunile si se astepta apasarea butonului).

2 → mainGame(scena care ruleaza jocul).

3 → endGame(scena care afiseaza detaliile de sfarsit).

startScene :

Cu ajutorul functiilor predefinite de biblioteca LCD-ului afisez 2 mesaje care contin titlul jocului si caracterul special pentru jucator. Actualizez variabila sceneManager ca sa treaca la urmatoarea scena.

mainMenu :

La fel ca la prima afisez 2 mesaje cu instructiunile pentru joc. Atunci cand pinul butonului citeste un 1 logic(butonul este apasat) atunci se actualizeaza variabila sceneManager ca sa trecem la urmatoarea scena.

mainGame :

Folosind variabila globala gameMap(care reprezinta o harta cu toate informatiile din joc), printez acest lucru pe lcd cu ajutorul functiei printMap. Dupa cu ajutorul functiei generateWall generez un perete aflat pe o linie random(aceasta valoare fiind calculata cu ajutorul functie newRandom) si plasez acel perete pe harta principala. Cu ajutorul functiei checkMoveShip verificam daca player-ul a cerut mutarea caracterului din joc( a apasat buton) si in caz ca da actualizez nou pozitie(incrementand cu 1) , facem %2 sa fim siguri ca nava nu este de pe harta si actualizam harta. La final moveWallToRight mutam toti peretii de pe harta cu o pozitie mai la dreapta si in acelasi timp verificam daca exista o coliziune cu nava noastra.

In cazul in care exista o coliziune atunci se actualizeaza sceneManager si se trece la scena urmatoare.

endGame :\\afisez instructiunile pe LCD si scorul final. In caz ca jucatorul vrea sa reincerce jocul(apasa pe buton) se actualizeaza sceneManager si se trece la scena mainGame.

Codul:

[DodgeTheWall.ino](#)

```
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 2, 3, 4, 5);

#define spaceship 0
#define spacewall 1
#define btn 9

byte ship[8] = {
  B00000,
  B00000,
  B00100,
  B11110,
  B00111,
  B11110,
  B00100,
  B00000
};

byte wall[8] = {
  B11111,
```

```
B11111,  
B11111,  
B11111,  
B11111,  
B11111,  
B11111,  
B11111  
};  
  
int gameMap[2][16] = {  
    1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
};  
  
int sceneManager = 0;  
float score = 0;  
  
void setup() {  
    // set up the LCD's number of columns and rows:  
    lcd.begin(16, 2);  
    pinMode(btn, INPUT);  
  
    // Print a message to the LCD.  
    Serial.begin(9600);  
    lcd.createChar(spaceship, ship);  
    lcd.createChar(spacewall, wall);  
}  
  
void sceneLoad(int sceneNumber){  
    switch (sceneNumber) {  
        case 0:  
            startScene();  
            break;  
        case 1:  
            mainMenu();  
            break;  
        case 2:  
            mainGame();  
            break;  
        case 3:  
            endGame();  
            break;  
        default:  
            // statements  
            break;  
    }  
}  
  
void startScene(){  
    lcd.clear();
```

```
    lcd.setCursor(0, 0);
    lcd.print("Dodge the wall");
    lcd.setCursor(6, 1);
    lcd.write(byte(spaceship));
    delay(2000);
    sceneManager = 1;
}

void mainMenu(){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Btn to play");
    lcd.setCursor(0, 1);
    lcd.print("Btn to jump");
    if(digitalRead(btn)){
        sceneManager = 2;
    }
    delay(200);
}

int yShip = 0;

unsigned long newrandom(unsigned long howsmall, unsigned long howbig)
{
    return howsmall + random() % (howbig - howsmall);
}

void checkMoveShip(){
    if(digitalRead(btn)){
        gameMap[yShip][0] = 0;
        yShip += 1;
        yShip %= 2;
        gameMap[yShip][0] = 1;
        delay(50);
    }
}

void generateWall(){
    if(newrandom(0, 5) == 0){
        int nr = newrandom(0,2);
        int nro = (nr + 1)%2;
        if(gameMap[nro][15] != 2 && gameMap[nro][14] != 2 && gameMap[nro][
13] != 2){
            gameMap[nr][15] = 2;
        }
    }
}

void printSpaceShip(int i, int j){
    lcd.setCursor(j, i);
```

```
    lcd.write(byte(spaceship));
}

void printWall(int i, int j){
    lcd.setCursor(j, i);
    lcd.write(byte(spacewall));
}

void printMap(){
    lcd.clear();
    for(int i = 0; i < 2;i++){
        for(int j = 0; j < 16;j++){
            switch (gameMap[i][j]){
                case 1:
                    printSpaceShip(i,j);
                    break;
                case 2:
                    printWall(i,j);
                    break;
                default:
                    break;
            }
        }
    }
}

void moveWallToRight(){
    for(int i = 0; i < 2;i++){
        for(int j = 0; j < 16;j++){
            if(gameMap[i][j] == 2){
                gameMap[i][j] = 0;
                if(gameMap[i][j-1] == 1){
                    sceneManager = 3;
                    lcd.clear();
                    lcd.print("Game over");
                    delay(2000);
                } else {
                    gameMap[i][j-1] = 2;
                }
            }
        }
    }
}

void mainGame(){
    score += 1;
    checkMoveShip();
    generateWall();
    printMap();
    moveWallToRight();
}
```

```
    delay(500);
}

void resetMap(){
    for(int i = 0; i < 2;i++){
        for(int j = 0;j < 16;j++){
            gameMap[i][j] = 0;
        }
    }
    gameMap[yShip][0] = 1;
}

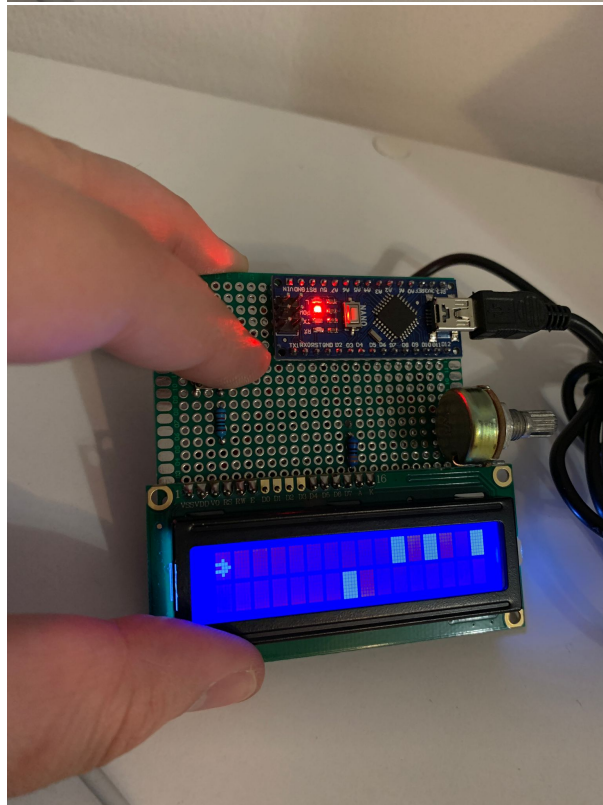
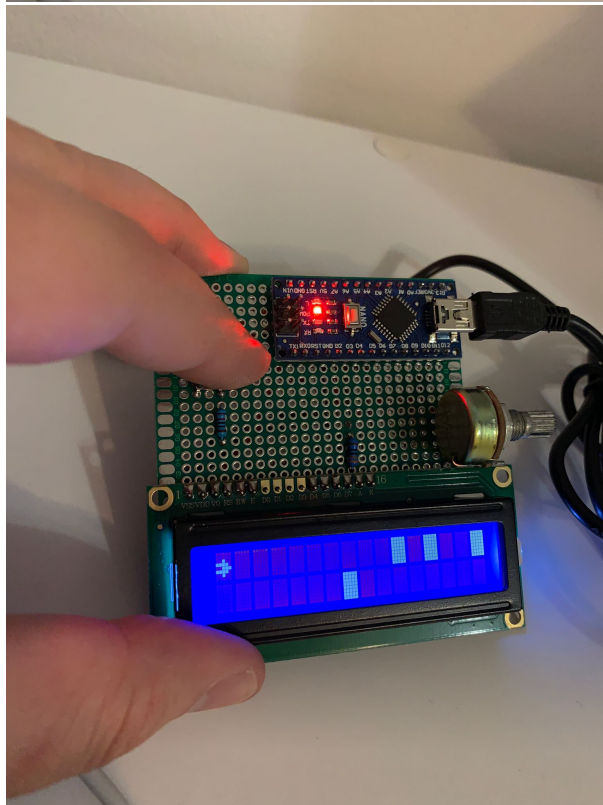
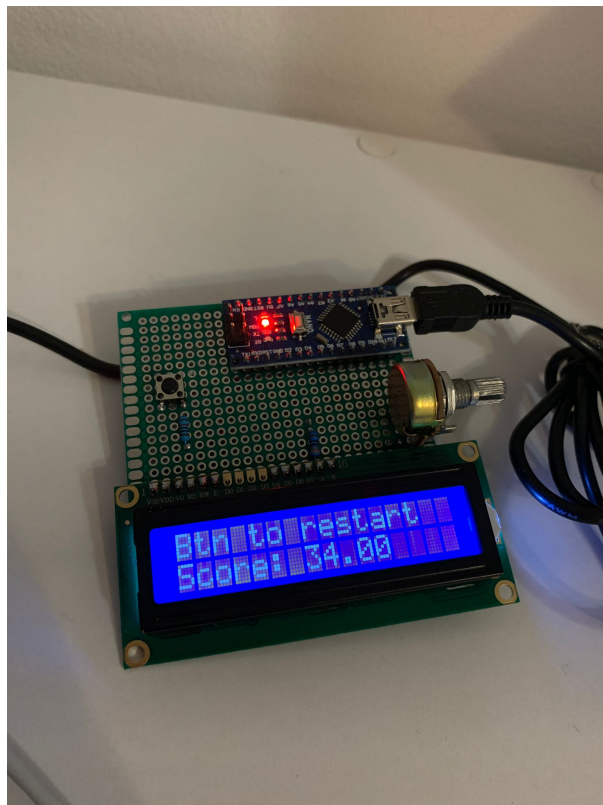
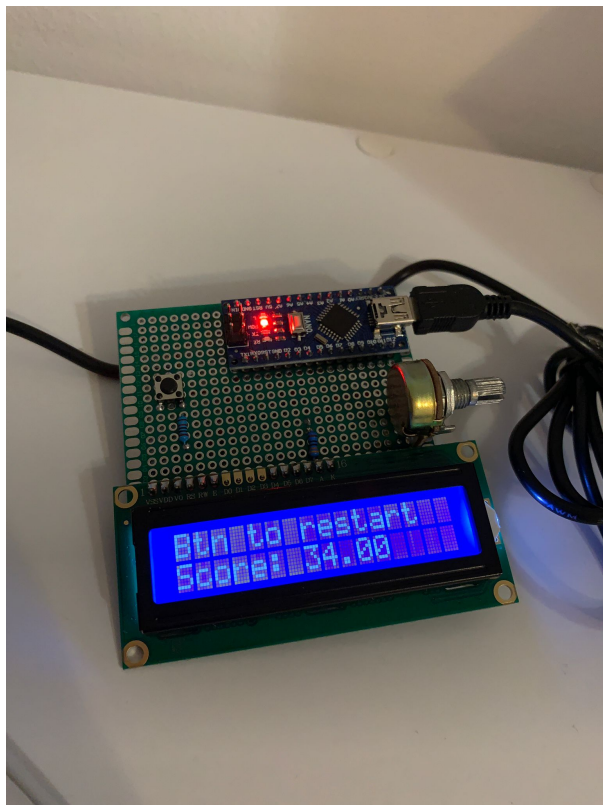
void endGame(){
    lcd.clear();
    lcd.print("Btn to restart");
    lcd.setCursor(0,1);
    lcd.print("Score: ");
    lcd.print(score);
    if(digitalRead(btn)){
        resetMap();
        sceneManager = 2;
        score = 0;
    }
    delay(200);
}

void loop() {
    sceneLoad(sceneManager);
}
```

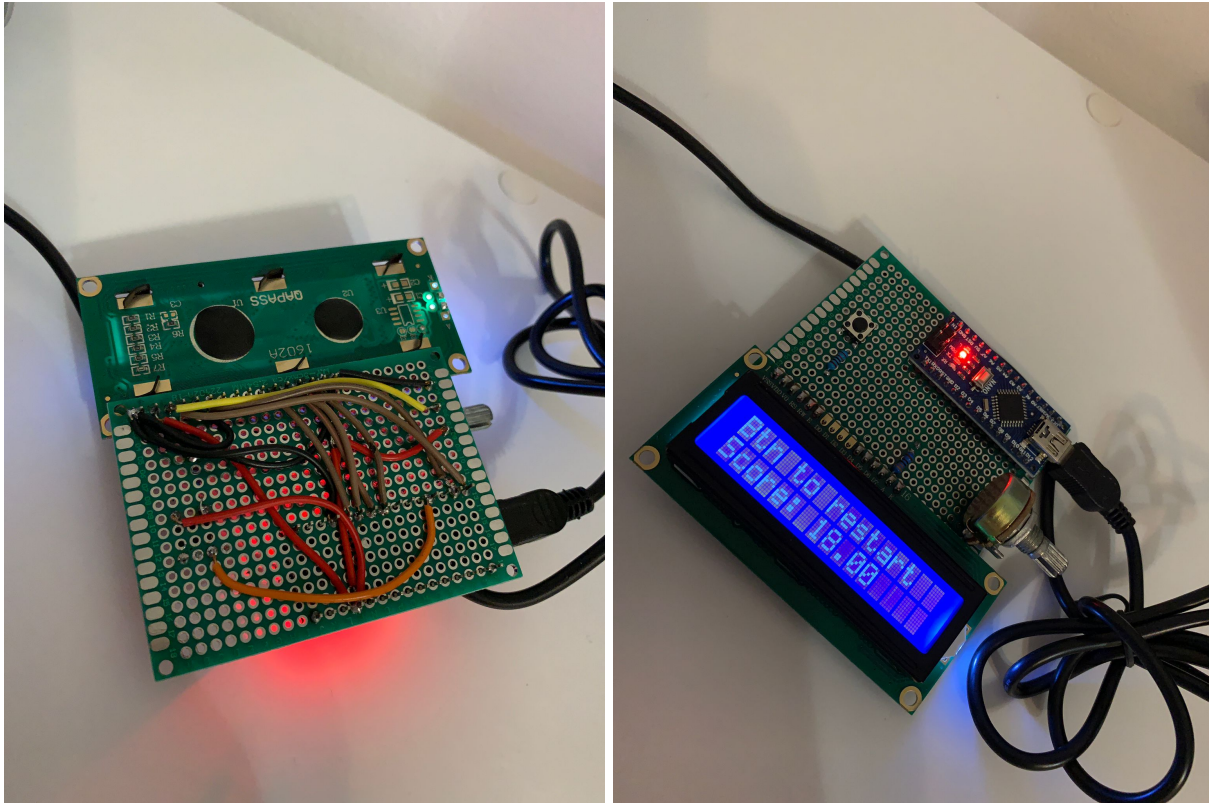
## Rezultate Obținute

Am obtinul un joc care mi se pare fun. Pe parcusul acestul drum am invatat foarte multe chestii. De la cum sa caut diferite informatii pe internet pana la lipirea unor fire.









## Download

Repo github: [https://github.com/fandreiPM/Dodge\\_the\\_wall](https://github.com/fandreiPM/Dodge_the_wall)

## Bibliografie/Resurse

Linkuri:

LCD:

<https://core-electronics.com.au/guides/use-lcd-arduino-uno/#:~:text=Insert%20your%20LCD%20screen%20into,ground%20your%20Backlight%20and%20LCD.>

Buton:

<https://create.arduino.cc/projecthub/muhammad-aqib/arduino-button-tutorial-using-arduino-digitalread-function-08adb5>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2022/agmocanu/dodgethewall>



Last update: **2022/05/27 21:00**

