

Cows And Bulls

Introducere

Proiectul ales este jocul Cows and Bulls un joc in care trebuie sa spargem un cod de 4 cifre distincte, cu ajutorul "Bulls" care ne spun cand am pus un numar corect si pe pozitia corecta pe cand, "Cows" ne spun ca avem numarul corect pe pozitia incorecta. Scopul proiectului este unul de distractie. Cred ca este util pentru ca este un mod usor de a mai scapa de stres printr-un joc simplu.

Descriere generală

Folosim tasta "*" de pe tastatura care va incepe un joc nou, ghicim folosindu-ne de tastele 1-9, dupa ce introducem 4 cifre apasam tasta "#" care va actiona ca tasta de input. Daca castigam buzzer-ul incepe sa cante si pe ecran va aparea un mesaj de felicitari altfel daca nu ghicim in 5 incercari am pierdut.



Hardware Design

- Arduino Nano
- Breadboard
- Matrix keypad 4*3
- Buzzer
- LCD

Software Design

Mediul de dezvoltare:

- Arduino IDE

Libraries used:

- Keypad.h - pentru lucrul cu keypad
- LiquidCrystal_I2C.h - pentru lucrul cu LCD

Organizarea codului:

Funcțiile următoare sunt pentru logica jocului:

```
void start()
{
    lcd.clear();
}
```

```
do
{
    set_random();

} while(check_duplicate(random_number)!=true);

    lcd.setCursor(0,0);
    lcd.print("Press 4 digits");
    tries=0;
    cows=0;
    bulls=0;
}
```

Funcția start() se folosește de toate celelalte funcții prezentate mai jos, care sunt logica jocului, se începe prin a curăța ecranul LCD, creând un număr random și verificând ca are cifrele distincte.

```
void set_random()
{
    for(x=0;x<=3;x++)
    {
        randomSeed(analogRead(analogRead(x)));

        randomNumber[x]=random(1,9);
    }

    random_number =
String(randomNumber[0])+String(randomNumber[1])+String(randomNumber[2])+String(randomNumber[3]);
}
```

Funcția set_random() creează un număr random de la 1-9 pe care îl copiează în variabila random_number pe care o comparăm cu numărul introdus de la keypad.

```
void enter()
{
    if(count==4)
    {
        k=String(generatedNumber_stored[0]);
        l=String(generatedNumber_stored[1]);
        m=String(generatedNumber_stored[2]);
        n=String(generatedNumber_stored[3]);

        player_number = k+l+m+n;

        if(check_duplicate(player_number))
        {
            check();
        }
    }
}
```

```

else
{
    lcd.setCursor(0,0);
    lcd.print("Incomplete");
}

count=0;
}

```

Funcția enter() verifică dacă tasta "#" a fost apăsată verificând dacă avem 4 cifre introduse care creează un număr cu cifre distincte.

```

void check()
{
    a1=String(randomNumber[0]);
    b1=String(randomNumber[1]);
    c1=String(randomNumber[2]);
    d1=String(randomNumber[3]);

    if(a1==k)    {bulls++;}

    if(b1==l)    {bulls++;}

    if(c1==m)    {bulls++;}

    if(d1==n)    {bulls++;}

    if(a1==l || a1==m || a1==n) { cows++;}
    if(b1==k || b1==m || b1==n) { cows++;}
    if(c1==k || c1==l || c1==n) { cows++;}
    if(d1==k || d1==l || d1==m) { cows++;}

    tries++;

    if(bulls==4)
    {
        lcd.clear();
        lcd.print("You won in: ");
        lcd.print(tries);
        lcd.setCursor(0,1);
        lcd.print( "tries");
        delay(2000);
        for (int thisNote = 0; thisNote < notes * 2; thisNote =
thisNote + 2) {
            divider = melody[thisNote + 1];
            if (divider > 0) {
                noteDuration = (wholenote) / divider;

```

```

        } else if (divider < 0) {
            noteDuration = (wholenote) / abs(divider);
            noteDuration *= 1.5; // increases the duration in
half for dotted notes
        }
        tone(buzzer, melody[thisNote], noteDuration*0.9);
        delay(noteDuration);
        noTone(buzzer);
    }
    start();
}
// Aici am introdus si cod-ul pentru buzzer care v-a incepe sa cante
cand jocul este castigat.
update_display();

if(tries==maxTries)
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.write("You lost, the");
    lcd.setCursor(0,1);
    lcd.print("number was: ");
    lcd.print(a1);
    lcd.print(b1);
    lcd.print(c1);
    lcd.print(d1);
}

count=0;
cows=0;
bulls=0;

}

```

Functia check() verifica daca numarul de bulls este 4, adica am castigat jocul, aici buzzer ul intervine cu un cantec care evidentiaza faptul ca am castigat. Numarul maxim de incercari fiind de 5.

```

void update_display()
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Cows:");
    lcd.print(cows);

    lcd.setCursor(8,0);
    lcd.print(" Bulls:");
    lcd.print(bulls);

    lcd.setCursor(9,1);
    lcd.print("Try:");
    lcd.print(tries);
}

```

```
        lcd.print("/5");

        lcd.setCursor(0,1);
        lcd.blink();
    }
```

Functia `update_display()` il ajuta pe jucator sa stie cate poziti cu numarul corect sunt (" Bulls:") si numarul corect dar pozitia incorecta (" Cows:"). LCD arata mereu in functie de numarul introdus de catre jucator in plus cate incercari mai are din 5.

Functile `setup()` si `loop()`:

```
void setup()
{
    Serial.begin(9600);
    lcd.begin();
    start();
}
```

Functia `setup()` porneste serial monitor-ul, ecranul si jocul Cows and Bulls prin functia `start()` explicata mai sus.

```
void loop()
{
    char key = customKeypad.getKey();
    if(key)
    {
        switch (key)
        {
            case '*':
                start(); //Starts a new game
                Serial.println(random_number); // implemented only to test the win,
                break;

            case '0':
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.print("Bye, the number ");
                lcd.setCursor(0,1);
                lcd.print("was: ");
                lcd.print(random_number);
                break;

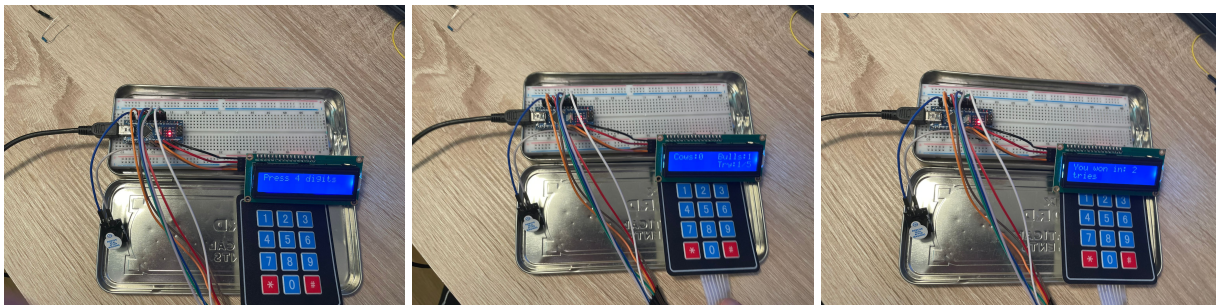
            case '#':
                enter(); //Enter
                break;

        }
    }
    if(key)
    {
```

```
generatedNumber_stored[count] = key;  
  
//  
if(key!='*' && key!='#' && key!='0')  
{  
    lcd.setCursor(0+count,1);  
    lcd.print(generatedNumber_stored[count]);  
  
    count++;  
}  
}  
}
```

Funcția loop() prin variabila key așteaptă de la keypad 4 cifre, dar în cazul în care avem tasta "*" jucătorul începe un nou joc cu alt număr, tasta "0" oprește jocul, și tasta "#" cum am evidențiat la funcția enter() este input-ul către joc.

Rezultate Obținute



Concluzii

Acest proiect mi s-a părut foarte interesant deoarece am înțeles mai bine cum să folosesc librării în Arduino, cum să fac legătura între partea hardware și partea software în cazul meu datele care le obțineam de la keypad să le introduc într-o variabilă cu care comparăm variabila random generată, după aceea să afișez aceste variabile pe LCD.

Pe viitor, voi încerca proiecte mai complexe folosindu-mă de mai multe componente hardware. Cu siguranță voi folosi aceste cunoștințe în a face proiecte for fun :).

Download

[proiect_cowsandbulls-chiritagabriel.rar](#)

Jurnal

Initial documentation (Introducere, Descriere generală, Hardware Design) - 7.05.2023

Final documentation (Software Design, Rezultate Obținute, Concluzii, Download, Bibliografie) - 27.05.2023

Bibliografie/Resurse

Resurse hardware:

<https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/>
<https://arduinogetstarted.com/tutorials/arduino-lcd-i2c>

Resurse software:

https://en.wikipedia.org/wiki/Bulls_and_Cows

Link: <https://ocw.cs.pub.ro/courses/pm/prj2023/tmiu/cowsandbulls>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2023/tmiu/cowsandbulls>



Last update: **2023/05/27 12:27**