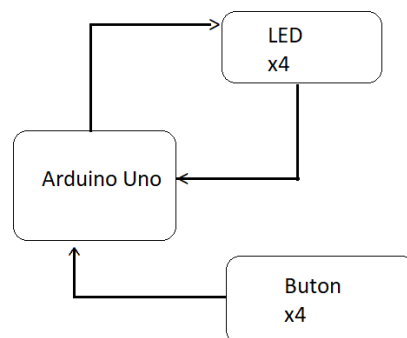


Introducere

Simon Says este un joc de memorie in care jucatorul trebuie sa repete o serie de instructiuni date de joc. Cu fiecare serie repetata corect jocul reincepte cu aceiasi serie la care adauga o noua instructiune. Cand jocul ajunge la numarul maxim de instructiuni sau cand o serie nu este repetata corect jocul se reseteaza.

Descriere

Sistemul este facut din 4 LED-uri de culori diferite si 4 butoane, toate conectate la un Arduino Board. Fiecarui buton ii corespunde un anumit LED, iar cand butonul este apasat LED-ul respectiv se aprinde. Dupa ce un numar de apasari echivalent cu lungimea sirului de instructiuni a fost introdus programul compara fiecare element din array-ul de instructiuni cu cel de pe aceiasi pozitie in array-ul de input. Daca o pereche de elemente sunt diferite compararea se opreste si jocul se reporneste, altfel jocul continua. Tiparirea in serial monitor nu e necesara dar ajuta sa vedem mai bine secventele de instructiuni/apasari de buton.



Hardware Design



Componente:

- Arduino Uno
- Breadboard

- 4 LED-uri in culori diferite(rosu,verde,galben,alb)
- 4 Butoane

Folosim LED-urile pentru a arata secventa ce trebuie introduse si butoanele pentru a o introduce, de asemenea LED-urile se aprind dupa apsarea butonului corespunzator pentru ca jucatorul sa obtina feedback ca circuitul a inregistrat apasarea de buton. Dupe ce LED-urile se aprind in secventa inregistrata intr-un array circuitul asteapta pana cand butoanele au fost apasate un numar egal cu lungimea array-ului.

Software Design

Code

```
#include <Arduino.h>

#define LEDR PD7
#define BUTTONR PD4
#define LEDG PD6
#define BUTTONG PD3
#define LEDY PD5
#define BUTTONY PD2
#define LEDW PB7
#define BUTTONW PD1

#define LED_TIME 2000
char simon[10];
char says[10];
int i=0;
int ct=0;
void setup() {

    DDRD |= (1 << LEDR);
    DDRD |= (1 << LEDG);
    DDRD |= (1 << LEDY);
    DDRB |= (1 << LEDW);

    DDRD &= ~(1 << BUTTONR);
    PORTD |= (1 << BUTTONR);

    DDRD &= ~(1 << BUTTONG);
```

```
    PORTD |= (1 << BUTTONG);

    DDRD &= ~(1 << BUTTONY);
    PORTD |= (1 << BUTTONY);

    DDRD &= ~(1 << BUTTONW);
    PORTD |= (1 << BUTTONW);
    Serial.begin(9600);
}

void loop() {
    while(i<2)
    {
        int start=random(4);
        if(start==0)
        {
            simon[i]='R';
            Serial.write("R");
        }
        if(start==1)
        {
            simon[i]='G';
            Serial.write("G");
        }
        if(start==2)
        {
            simon[i]='Y';
            Serial.write("Y");
        }
        if(start==3)
        {
            simon[i]='W';
            Serial.write("W");
        }
        i++;
    }
    int add=random(4);
    if(add==0)
    {
        simon[i]='R';
        Serial.write("R");
    }
    if(add==1)
    {
        simon[i]='G';
        Serial.write("G");
    }
    if(add==2)
    {
        simon[i]='Y';
```

```
Serial.write("Y");

}
if(add==3)
{
    simon[i]='W';
    Serial.write("W");
}
Serial.write('\n');
for(int j=0;j++;j<=i)
{
    if (simon[i]=='R'){
        PORTD |= (1 << LEDR);
        delay(LED_TIME);
        PORTD &= ~(1 << LEDR);
    }
    if (simon[i]=='G'){
        PORTD |= (1 << LEDG);
        delay(LED_TIME);
        PORTD &= ~(1 << LEDG);
    }
    if (simon[i]=='Y'){
        PORTD |= (1 << LEDY);
        delay(LED_TIME);
        PORTD &= ~(1 << LEDY);
    }
    if (simon[i]=='W'){
        PORTB |= (1 << LEDW);
        delay(LED_TIME);
        PORTB &= ~(1 << LEDW);
    }
}
while(ct<=i){

    if ((PIND & (1 << BUTTONR)) == 0) {
        PORTD |= (1 << LEDR);
        delay(LED_TIME);
        PORTD &= ~(1 << LEDR);
        says[ct]='R';
        Serial.write("R");
        ct++;
    }
    else if ((PIND & (1 << BUTTONG)) == 0) {
        PORTD |= (1 << LEDG);
        delay(LED_TIME);
        PORTD &= ~(1 << LEDG);
        says[ct]='G';
        Serial.write("G");
        ct++;
    }
}
```

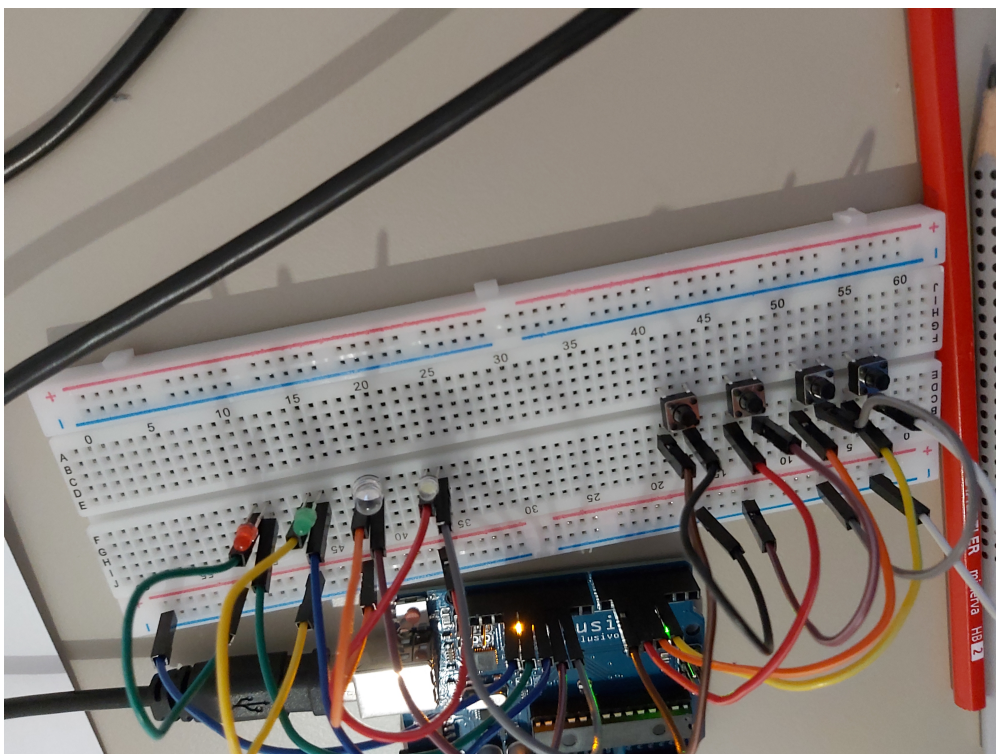
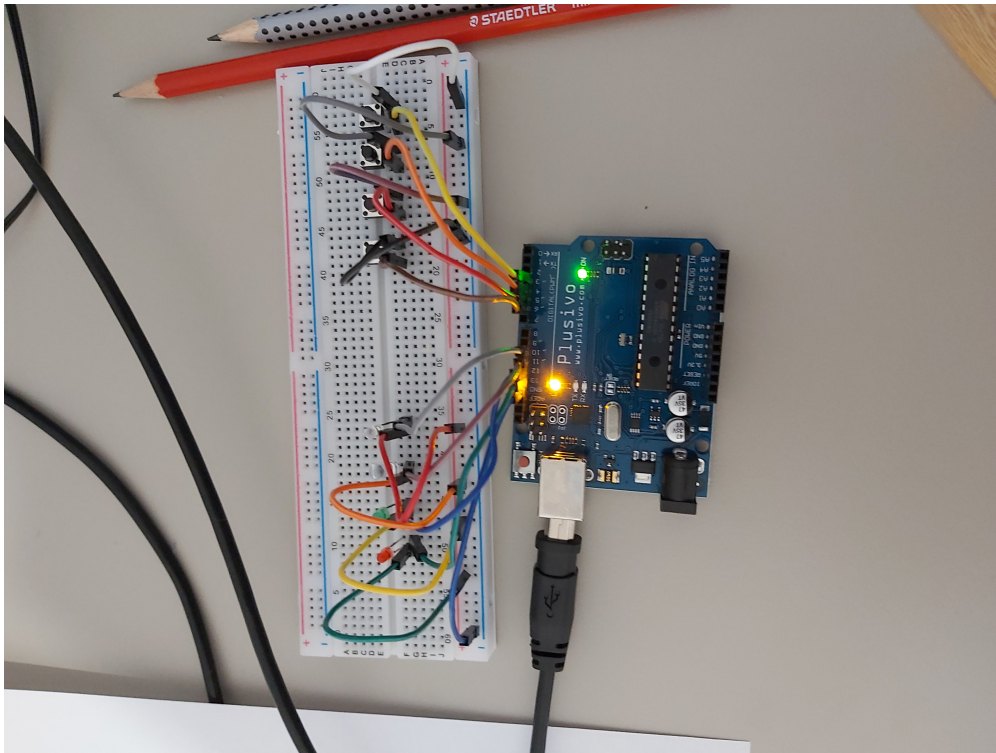
```
    else if ((PIND & (1 << BUTTONY)) == 0) {
        PORTD |= (1 << LEDY);
        delay(LED_TIME);
        PORTD &= ~(1 << LEDY);
        says[ct]='Y';
        Serial.write("Y");
        ct++;
    }
    else if ((PIND & (1 << BUTTONW)) == 0) {
        PORTB |= (1 << LEDW);
        delay(LED_TIME);
        PORTB &= ~(1 << LEDW);
        says[ct]='W';
        Serial.write("W");
        ct++;
    }
    delay(5000);

}
Serial.write('\n');
int ok=1;
for (int j=0;j<=i;j++)
{
    if(simon[j]!=says[j])
    {
        ok=0;
        break;
    }
}
ct=0;
if(ok==1){

    i++;
}
if(i==10||ok==0){
    i=0;
    PORTB |= (1 << LEDG);
    PORTB |= (1 << LEDY);
    PORTB |= (1 << LEDR);
    PORTB |= (1 << LEDW);
    delay(LED_TIME);
    PORTB &= ~(1 << LEDW);
    PORTB &= ~(1 << LEDR);
    PORTB &= ~(1 << LEDY);
    PORTB &= ~(1 << LEDG);
}
}
```

Rezultate Obtinute

Datorita unor erori in setarea programului arduino pe laptop nu am putut verifica pe hardware codul deci a fost necesar sa il simulez pe tinkercad. Acolo partea de initializare a secventei merge, dar primele 3 instructiunii sunt mereu WGG nu stiu daca este de la cod sau daca pe simulare random incepe mereu cu aceasi secventa. In Tinkercad nu pare sa existe o animatie pentru aprinderea unui LED. La partea de introducere a secventei prin butoane indiferent de ce buton ai apasat rezultatul este Y si mai mult while-ul se opreste chiar daca nu se incrementeaza contorul, nu stiu ce ar putea cauza asta.



Concluzii

Dificultatile tehnice in asamblarea codului au creat o inabilitate de a verifica cu acuratete daca problema este din cod sau din programul folosit, totusi erorile observate arata o intercalare a mai multor semnale pe aceeasi banda nu o eroare a codului propriu-zis, iar asta necesita schimbari minime pentru a fi rezolvate.

Download

Archiva cu cod, scheme, imagini: [proiect_aurdino.rar](#)

Journal

- 2-8 mai-alegere tema proiect
- 19 mai-comandă piese
- 23 mai-6 iunie-realizare proiect
- 6 iunie-realizare documentaie

Bibliografie/Resurse

Resurse Hardware:

<https://www.optimusdigital.ro/ro/>

Resurse Software:

<https://www.tinkercad.com/dashboard>

[Export to PDF](#)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2022/dbrezeanu/simon_says



Last update: **2022/06/06 20:22**