



Ação de Formação:
Programação com Arduinos



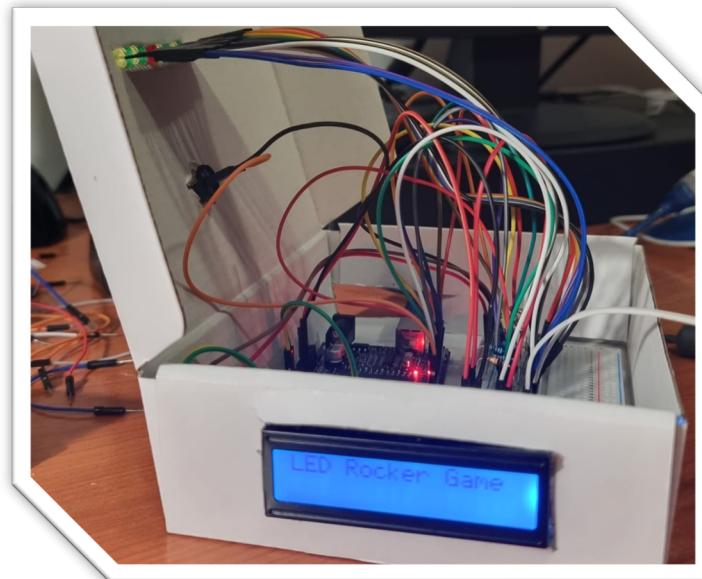
09/01/2024 a 12/03/2024

Formador: Luís Fernandes

Ano Letivo 2023/2024

Projeto

LED Rocker Game



Formando: Luís Baptista

Índice

Proposta de Projeto	3
Material utilizado.....	4
Desenvolvimento do projeto	5
Parte 1 – Montagem do circuito e código inicial	5
Imagens do protótipo e dos testes – Parte 1	6
Funcionamento (Monitor Série).....	7
Código da Parte 1	8
Vídeo da parte 1	9
Parte 2 – Adicionar LCD e Buzzer.....	10
Imagens do protótipo e dos testes – Parte 2	10
Código da Parte 2	11
Vídeo da parte 2	14
Parte 3 – Aperfeiçoamento e finalização do código	15
Código da Parte 3 (Código Final do Jogo)	15
Vídeos da parte 3	18
Parte 4 – Elaboração da caixa	19
Resultados / Melhorias futuras.....	26

Proposta de Projeto

Formando

LUÍS ALBINO DE CASTRO BAPTISTA

Designação do Projeto

Arduino LED Rocker Game

Enquadramento

Como projeto final, pretende-se que seja desenvolvido um projeto, utilizando o microcontrolador Arduino, que evidencie as competências adquiridas ao longo da ação de formação.

A construção do jogo LED Rocker Game, com Arduino e componentes eletrónicos, enquadra-se nos objetivos do projeto final, tendo sido escolhida a partir das propostas apresentadas pelo formador. Será utilizada uma placa Arduino UNO e diversos componentes para realizar o projeto.

Para a apresentação do projeto final será efetuada a gravação de um pequeno vídeo ou uma apresentação automática que incluirá:

- Designação do projeto
- Objetivos do projeto
- Componentes e materiais utilizados
- Esquemas de ligação
- Fases de construção
- Demonstração

Objetivos

O objetivo deste trabalho que proponho fazer é a criação do jogo LED ROCKER GAME, utilizando a placa eletrónica ARDUINO e os diversos componentes eletrónicos necessários para a sua implementação física.

O jogo tem um conjunto de LEDs que acendem sequencialmente num sentido e, depois, no sentido contrário, em ciclo até que o jogador prima o botão de pressão. Nesse momento o código tem de testar se o LED que estava acesso corresponde ao LED do meio. Se for o caso, o jogador ganha pontos (ou vidas). Caso o LED central não esteja ligado no momento de premir o botão, então o jogador perde pontos (vidas).

O jogo termina se o número de vidas chegar a zero e o jogador falhar de novo.

Na implementação pretendo utilizar um display LCD para ir mostrando o desenrolar do jogo.

Irei, ainda, fazer o código em linguagem C e integrar toda a eletrónica numa caixa, apresentando um trabalho completo em forma de aparelho eletrónico.

Recursos Necessários

Para realizar este projeto necessitarei dos seguintes componentes:

- 1 Placa Arduino Uno
- 1 Breadboard
- 9 LEDs (1 vermelho; 8 de outra cor)
- 1 botão de pressão
- 1 LCD panel
- 1 speaker ou buzzer passivo
- Vários fios (jumper wires)
- 10 resistências de 220 ohm
- 1 cabo de energia/USB

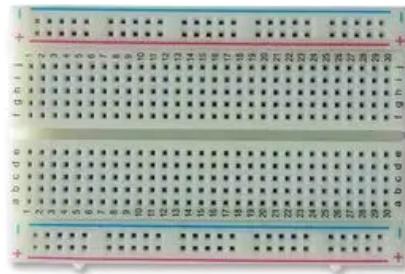
Material utilizado

O material utilizado para este projeto foi:

1 Placa Arduino Uno



1 Breadboard



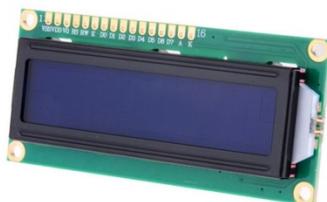
9 LEDs (1 vermelho; 4 verdes; 4 amarelos)



1 botão de pressão



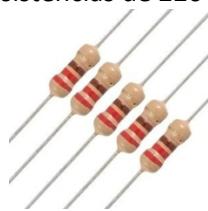
1 LCD panel



1 buzzer passivo



9 resistências de 220 ohm



1 resistência de 10 kohm



1 cabo de energia/USB



Vários fios macho-macho e macho-fêmea



Desenvolvimento do projeto

Para desenvolver o projeto, dividi o trabalho em 4 partes essenciais, detalhadas à frente:

- Na **primeira parte** fiz a montagem do circuito base e escrevi o código inicial, fazendo os testes de funcionamento. Este circuito inicial não tinha, ainda, o LCD nem o buzzer. Pretendi testar o código para verificar se o jogo funcionava. Utilizei para isso a comunicação com o Monitor Série.
- Após verificar que o jogo estava a funcionar, numa **segunda parte** adicionei o display LCD para as mensagens serem vistas através deste e um *buzzer* para produzir sons de ganhar e perder vidas, bem como o de perder o jogo. Estes dois elementos permitem dar maior realismo ao jogo através da interatividade que adicionam.
- Após verificar que tudo estava funcional com o jogo e a comunicação com o LCD, na **terceira parte** aperfeiçoei o código mostrado no display durante o jogo bem como os sons emitidos pelo *buzzer* e a sua duração. Ajustei, ainda, os *timings* de espera após carregar no botão de pressão e após o fim do jogo, bem como a velocidade dos leds após cada tentativa. No final do jogo, o utilizador tem de esperar 10 segundos e tudo recomeça para um novo jogo / novo jogador.
- A **quarta parte** foi para criar uma caixa de embalagem do jogo, que tentei que ficasse minimamente atrativa. Para tal utilizei uma caixa de cartão de um equipamento comprado, fiz os furos e cortes necessários e ajustei o circuito lá dentro.

Parte 1 – Montagem do circuito e código inicial

Na primeira parte fiz a montagem do circuito essencial para verificar e testar o funcionamento do jogo com o código que produzi.

Utilizei a placa *Arduino Uno*, a *breadboard*, um led vermelho para indicar o LED principal, quatro LEDs amarelos e quatro verdes (tive que utilizar cores diferentes porque não tinha oito LEDs de uma só cor), um botão de pressão e as resistências necessárias para ligar os LEDs e o botão (220 ohm e 10 kohm, respetivamente). Liguei os fios (*jumper wires*) entre os pontos da *breadboard* e da placa *Arduino*.

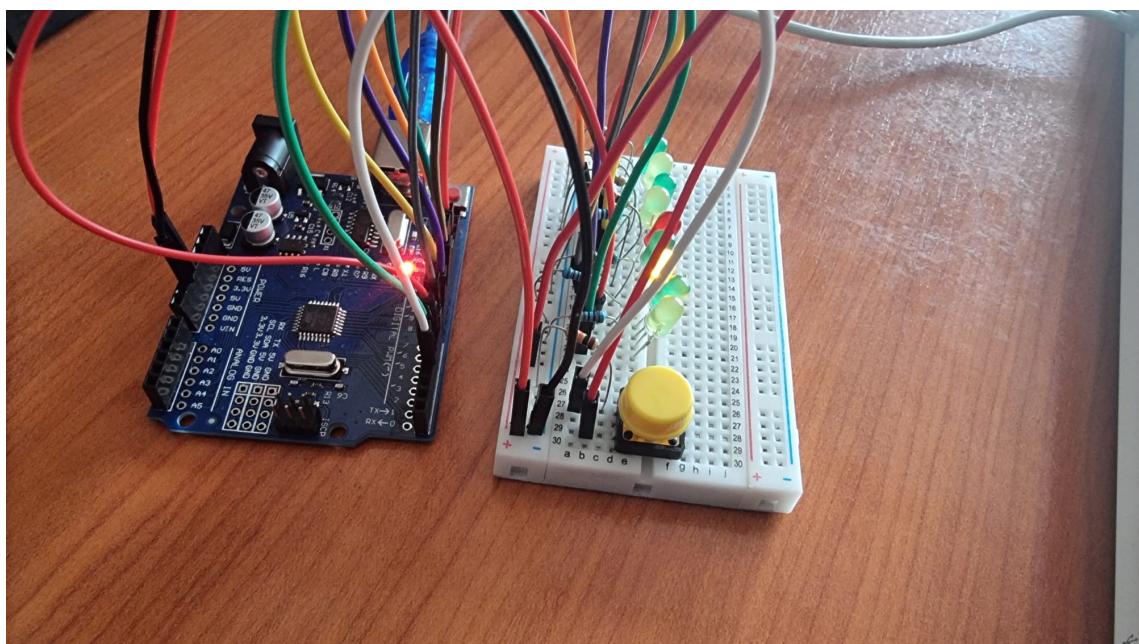
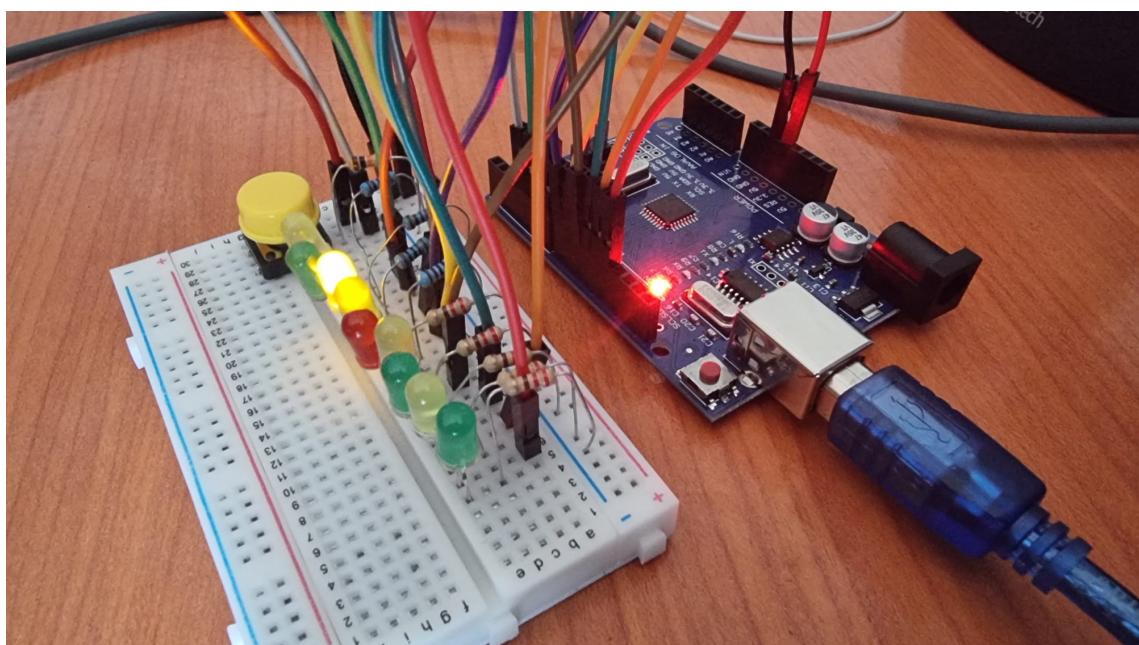
Utilizei as portas 5 a 13 para ligar os LEDs e a porta 4 para o botão.

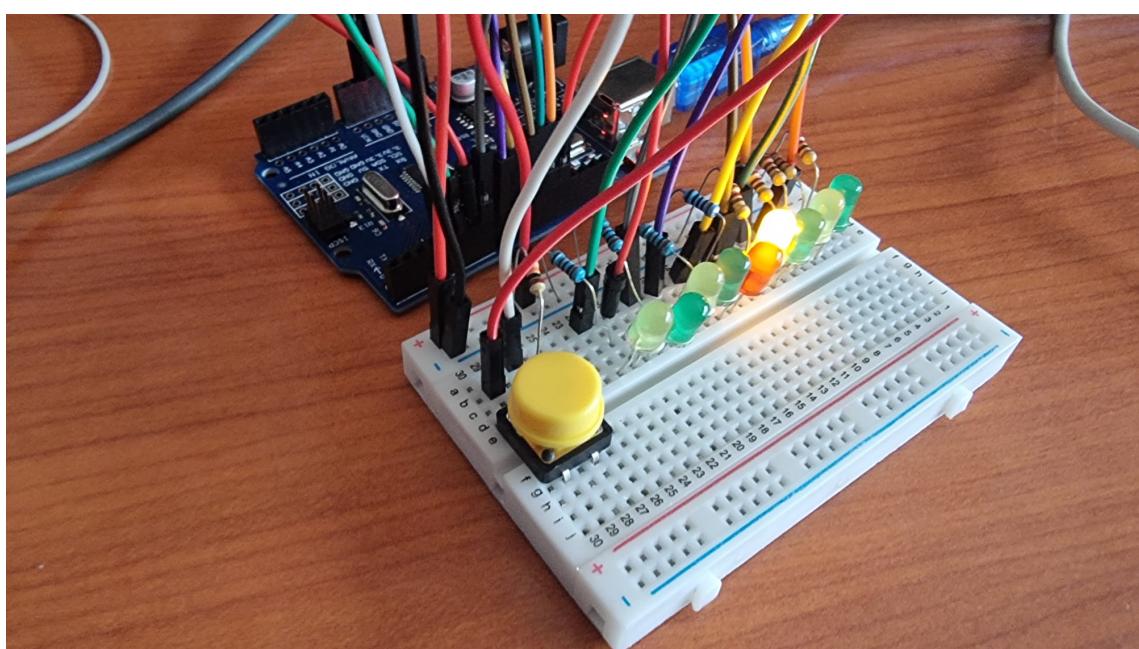
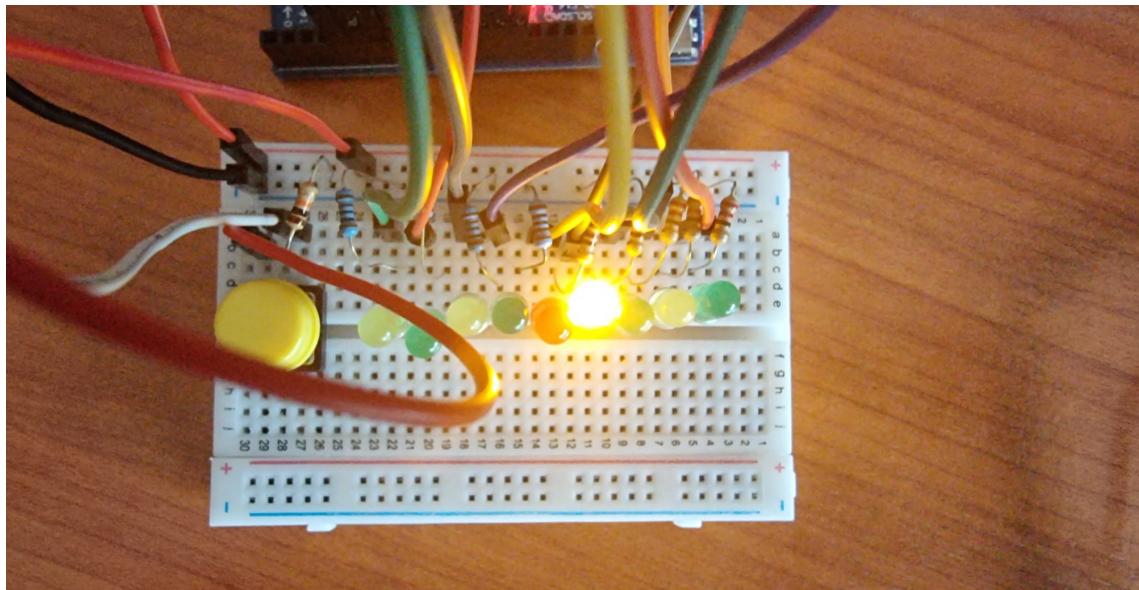
O LED do meio (vermelho) é onde se ganham vidas caso se consiga carregar no botão de pressão quando este estiver ligado. Se carregar no botão quando estiver outro LED ligado, o jogador perde vidas.

Depois de concluir este protótipo inicial, fiz os testes ao código com o *Serial Monitor*, verificando que estava tudo funcional.

Imagens do protótipo e dos testes – Parte 1

Nesta primeira parte, os resultados obtidos para testar foram os seguintes:





Funcionamento (Monitor Série)

```
Output Serial Monitor ×  
Message (Enter to send message to 'Arduino Uno' on '/dev/cu.  
15:16:57.209 -> Premiu led: 3  
15:16:57.341 -> Errou, nível menor  
15:16:57.341 -> Vidas = 2  
15:17:04.656 -> Premiu led: 8  
15:17:04.755 -> Errou, nível menor  
15:17:04.755 -> Vidas = 1  
15:17:11.639 -> Premiu led: 4  
15:17:11.737 -> Errou, nível menor  
15:17:11.737 -> Vidas = 0  
15:17:11.770 -> Fim do jogo  
15:17:21.736 -> Novo jogo, vidas 3
```

Código da Parte 1

```
const int ledPin[] = { 5, 6, 7, 8, 9, 10, 11, 12, 13 };

#define btnPin 4

int i, botao, pinAtual;

int wait = 500;
int vidas = 3;

void setup() {
    for (i = 0; i < 9; i++) {
        pinMode(ledPin[i], OUTPUT);
    }
    pinMode(btnPin, INPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    Serial.begin(9600);

    for (i = 0; i < 8; i++) {
        digitalWrite(ledPin[i], HIGH);
        pinAtual = i;
        //Serial.println(pinAtual);
        btnPress();
        delay(wait);
        digitalWrite(ledPin[i], LOW);
        btnPress();
        delay(wait);
    }

    for (i = 8; i > 0; i--) {
        digitalWrite(ledPin[i], HIGH);
        pinAtual = i;
        //Serial.println(pinAtual);
        btnPress();
        delay(wait);
        digitalWrite(ledPin[i], LOW);
        btnPress();
        delay(wait);
    }
}

void btnPress() {
    botao = digitalRead(btnPin);
    if (botao == 1) {
```

```

//wait -= 50;
//desligarPin();
digitalWrite(ledPin[pinAtual], HIGH);
Serial.print("Premiu led: ");
Serial.println(pinAtual + 1);
delay(100);
digitalWrite(ledPin[pinAtual], HIGH);
if (vidasTest() == 0)
{
    Serial.println("Fim do jogo");
    desligarPin();
    vidas = 3;
    delay(10000);
    Serial.println("Novo jogo, vidas 3");
}
wait /= 1.5;
delay(4000);

}

void desligarPin()
{
    for(i=0; i<9; i++) digitalWrite(ledPin[i], LOW);
}

int vidasTest()
{
    if ((pinAtual + 1) == 5)
    {
        Serial.println("Acertou, nível seguinte");
        vidas += 1;
        Serial.print("Vidas = ");
        Serial.println(vidas);

    }
    else {
        Serial.println("Errou, nível menor");
        vidas -= 1;
        Serial.print("Vidas = ");
        Serial.println(vidas);

    }
    return vidas;
}

```

Vídeo da parte 1

No link seguinte pode ser visualizado um vídeo do protótipo nesta fase:

https://drive.google.com/file/d/1D4o-v1-xKv6Ywo7fNECZguARJQ8X6nP1/view?usp=share_link

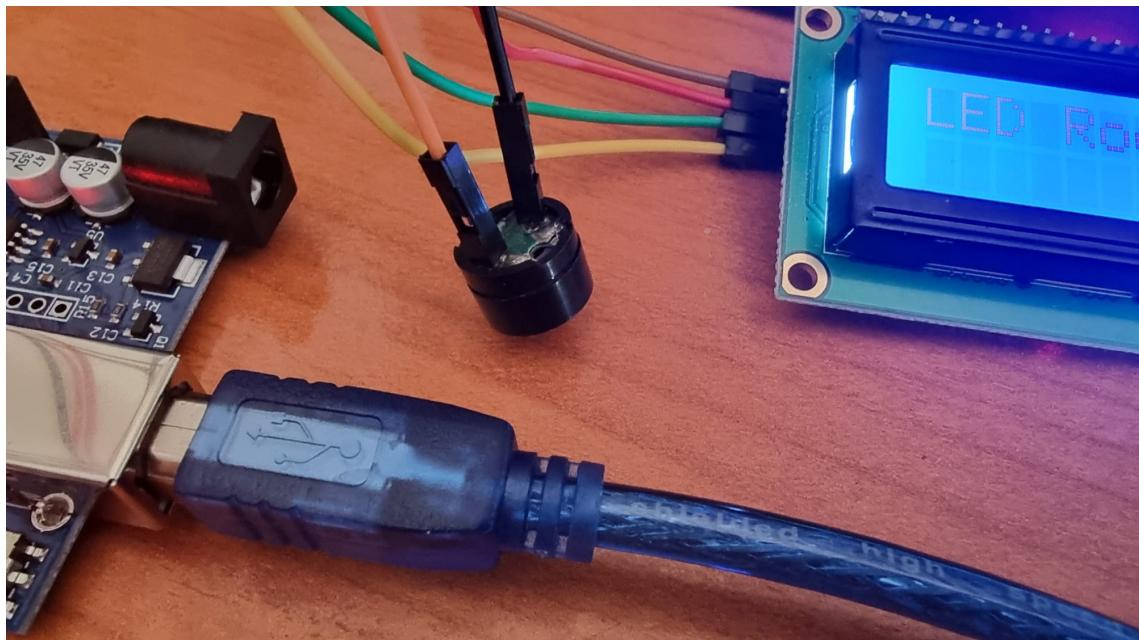
Parte 2 – Adicionar LCD e Buzzer

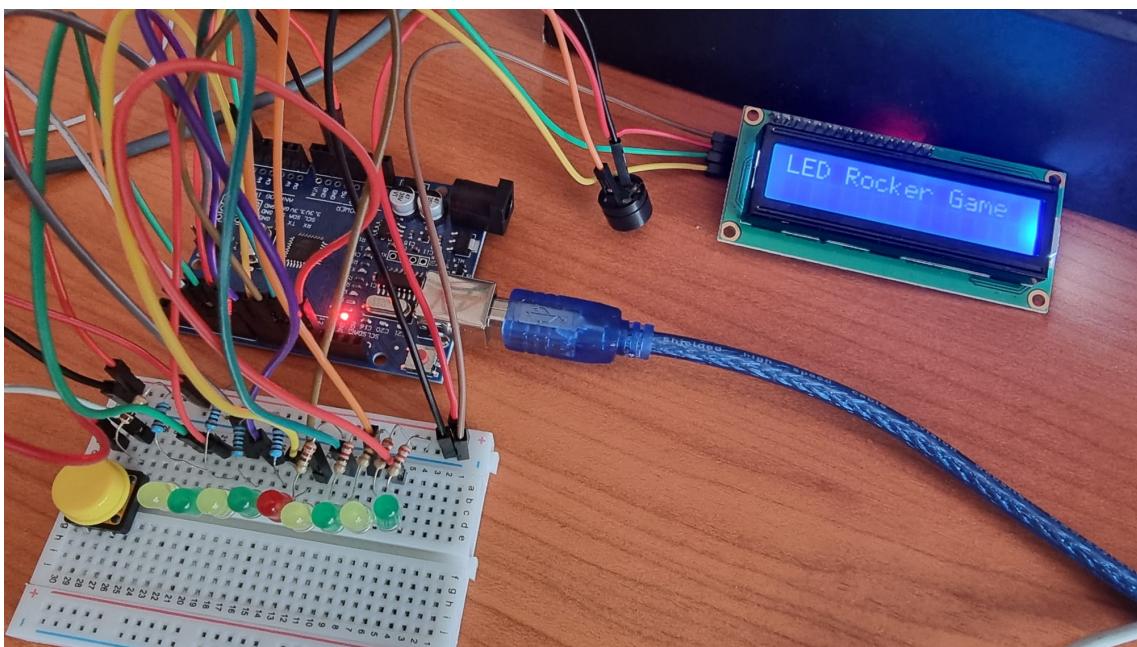
Na segunda parte do desenvolvimento do projeto adicionei um display LCD para criar interatividade com o utilizador (apresenta as informações do jogo – ganhar, perder, número de vidas, novo jogo) e um *buzzer* para indicar com um som se ganhou, se perdeu e se o jogo terminou.

Utilizei mais fios (*jumper wires*) entre os pontos da *breadboard*, do *buzzer*, do LCD e da placa *Arduino*. Adicionei fios macho – fêmea, para fazer as ligações dos pinos do *buzzer* e do LCD às placas.

Alterei o código (adicionei código, essencialmente) para mostrar informação no LCD e para fazer o *buzzer* emitir sons.

Imagens do protótipo e dos testes – Parte 2





Código da Parte 2

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int ledPin[] = { 5, 6, 7, 8, 9, 10, 11, 12, 13 };

#define buzzerPin 3

#define btnPin 4
```

```

int i, botao, pinAtual;

int wait = 500;
int vidas = 3;

void setup() {
    for (i = 0; i < 9; i++) {
        pinMode(ledPin[i], OUTPUT);
    }
    pinMode(btnPin, INPUT);
    pinMode(buzzerPin, OUTPUT);
    lcd.init();
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
    //Serial.begin(9600);

    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("LED Rocker Game");
    lcd.setCursor(0, 1);

    for (i = 0; i < 8; i++) {
        digitalWrite(ledPin[i], HIGH);
        pinAtual = i;
        //Serial.println(pinAtual);
        btnPress();
        delay(wait);
        digitalWrite(ledPin[i], LOW);
        btnPress();
        delay(wait);
    }

    for (i = 8; i > 0; i--) {
        digitalWrite(ledPin[i], HIGH);
        pinAtual = i;
        //Serial.println(pinAtual);
        btnPress();
        delay(wait);
        digitalWrite(ledPin[i], LOW);
        btnPress();
        delay(wait);
    }
}

void btnPress() {

```

```

botaao = digitalRead(btnPin);
if (botaao == 1) {
    //wait -= 50;
    //desligarPin();
    digitalWrite(ledPin[pinAtual], HIGH);
    Serial.print("Premiu led: ");
    lcd.print("Premiu led: ");
    Serial.println(pinAtual + 1);
    lcd.print(pinAtual + 1);
    delay(100);
    digitalWrite(ledPin[pinAtual], HIGH);
    wait /= 1.5;
    if (vidasTest() == 0)
    {
        Serial.println("Fim do jogo");
        lcd.print("Fim do jogo");
        desligarPin();
        tone(buzzerPin, 349);
        delay(50);
        tone(buzzerPin, 330);
        delay(50);
        tone(buzzerPin, 294);
        delay(50);
        tone(buzzerPin, 262);
        delay(200);
        noTone(buzzerPin);
        vidas = 3;
        delay(10000);
        wait = 500;
        Serial.println("Novo jogo, vidas 3");
        lcd.print("Novo jogo, vidas 3");
    }
    //wait /= 1.5;
    delay(4000);
}

void desligarPin()
{
    for(i=0; i<9; i++) digitalWrite(ledPin[i], LOW);
}

int vidasTest()
{
    if ((pinAtual + 1) == 5)
    {
        tone(buzzerPin, 349);
        delay(50);
        noTone(buzzerPin);
        Serial.println("Acertou, nível seguinte");
    }
}

```

```

lcd.print("Acertou, nível seguinte");
vidas += 1;
Serial.print("Vidas = ");
lcd.print("Vidas = ");
Serial.println(vidas);
lcd.print(vidas);

}

else {
    tone(buzzerPin, 262);
    delay(200);
    noTone(buzzerPin);
    Serial.println("Errou, nível menor");
    lcd.print("Errou, nível menor");
    vidas -= 1;
    Serial.print("Vidas = ");
    lcd.print("Vidas = ");
    Serial.println(vidas);
    lcd.print(vidas);
}

return vidas;
}

```

Vídeo da parte 2

No link seguinte pode ser visualizado um vídeo do protótipo nesta fase:

https://drive.google.com/file/d/1QyvJ1wAAsgCx2bWchFzo5lu02ds7w6Pe/view?usp=share_link

Parte 3 – Aperfeiçoamento e finalização do código

Na terceira parte do desenvolvimento do projeto dediquei-me a fazer testes ao jogo e aperfeiçoei o código para os timings de cada etapa serem os melhores.

Em termos de LCD e da informação apresentada, melhorei o código para o mostrador dar a informação completa, em duas linhas, sem cortes.

No *buzzer*, para o som de perder, utilizei um som mais grave e prolongado. Para ganhar, utilizei um som mais agudo e rápido. No caso de perder as vidas todas (fim do jogo), utilizei um conjunto de 4 sons (usados nas sessões da formação), decrescendo as tonalidades.

O código ficou concluído, o jogo ficou totalmente funcional.

A execução do jogo é feita da seguinte forma:

1. Ligando o aparelho, aparece a mensagem LED Rocker Game, no LCD.
2. A velocidade de acender os LEDs é relativamente lenta (meio segundo, no início).
3. Se pressionar o botão e acertar no LED do meio (vermelho), ganha uma vida e mostra essa informação no LCD. Emite um som agudo e rápido.
4. Se pressionar e não acertar no LED do meio, perde uma vida e mostra a informação no LCD. Indica, sempre, o número de vidas que tem.
5. Depois de pressionar o botão e ganhar ou perder, a velocidade de acender os LEDs aumenta sempre 1.5 vezes. Ganhando muitas vidas, vai chegar a um ponto em que a velocidade é tal que não conseguirá acertar mais e começará a perder.
6. O jogo vai terminar quando não tiver mais vidas (vidas = 0). Nesse caso, mostra a informação no LCD e emite um conjunto de quatro sons, decrescentes. Utilizei as tonalidades do trabalho realizado na atividade 8, feita na sessão 4.

```
int tones[] = {262, 294, 330, 349};
```

Código da Parte 3 (Código Final do Jogo)

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int ledPin[] = { 5, 6, 7, 8, 9, 10, 11, 12, 13 };

#define buzzerPin 3
```

```

#define btnPin 4

int i, botao, pinAtual;

int wait = 500;
int vidas = 3;

void setup() {
    for (i = 0; i < 9; i++) {
        pinMode(ledPin[i], OUTPUT);
    }
    pinMode(btnPin, INPUT);
    pinMode(buzzerPin, OUTPUT);
    lcd.init();
    Serial.begin(9600);
}

void loop() {
    // put your main code here, to run repeatedly:
    //Serial.begin(9600);

    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("LED Rocker Game");
    lcd.setCursor(0, 1);

    for (i = 0; i < 8; i++) {
        digitalWrite(ledPin[i], HIGH);
        pinAtual = i;
        //Serial.println(pinAtual);
        btnPress();
        delay(wait);
        digitalWrite(ledPin[i], LOW);
        btnPress();
        delay(wait);
    }

    for (i = 8; i > 0; i--) {
        digitalWrite(ledPin[i], HIGH);
        pinAtual = i;
        //Serial.println(pinAtual);
        btnPress();
        delay(wait);
        digitalWrite(ledPin[i], LOW);
        btnPress();
        delay(wait);
    }
}

```

```

}

void btnPress() {
    botaao = digitalRead(btnPin);
    if (botaao == 1) {
        //wait -= 50;
        //desligarPin();
        digitalWrite(ledPin[pinAtual], HIGH);
        Serial.print("Premiu led: ");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Premiu led: ");
        Serial.println(pinAtual + 1);
        lcd.setCursor(0, 1);
        lcd.print(pinAtual + 1);
        delay(100);
        digitalWrite(ledPin[pinAtual], HIGH);
    }
    //0s vídeos a executar tinham este led ligado, para ver que estava a detetar
    corretamente – na realidade devia ser LOW

    wait /= 1.5;
    if (vidasTest() == 0)
    {
        Serial.println("Fim do jogo");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Fim do jogo");
        desligarPin();
        tone(buzzerPin, 349);
        delay(50);
        tone(buzzerPin, 330);
        delay(50);
        tone(buzzerPin, 294);
        delay(50);
        tone(buzzerPin, 262);
        delay(200);
        noTone(buzzerPin);
        vidas = 3;
        delay(10000);
        wait = 500;
        Serial.println("Novo jogo, vidas 3");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Novo jogo");
        lcd.setCursor(0, 1);
        lcd.print("3 vidas");
    }
    //wait /= 1.5;
    delay(4000);

}

```

```

}

void desligarPin()
{
    for(i=0; i<9; i++) digitalWrite(ledPin[i], LOW);
}

int vidasTest()
{
    if ((pinAtual + 1) == 5)
    {
        tone(buzzerPin, 349);
        delay(50);
        noTone(buzzerPin);
        Serial.println("Acertou, nível+");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Acertou, nível+");

        vidas += 1;
        Serial.print("Vidas = ");
        //lcd.clear();
        lcd.setCursor(0, 1);
        lcd.print("Vidas = ");
        Serial.println(vidas);
        lcd.print(vidas);

    }
    else {
        tone(buzzerPin, 262);
        delay(200);
        noTone(buzzerPin);
        Serial.println("Errou, nível-");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Errou, nível-");
        vidas -= 1;
        Serial.print("Vidas = ");
        //lcd.clear();
        lcd.setCursor(0, 1);
        lcd.print("Vidas = ");
        Serial.println(vidas);
        lcd.print(vidas);
    }
    return vidas;
}

```

Vídeo da parte 3

No link seguinte pode ser visualizado um vídeo do protótipo nesta fase:

https://drive.google.com/file/d/1roU_V-vBGDymkACBvIcZFr80bw6ruRH4/view?usp=share_link

Parte 4 – Elaboração da caixa

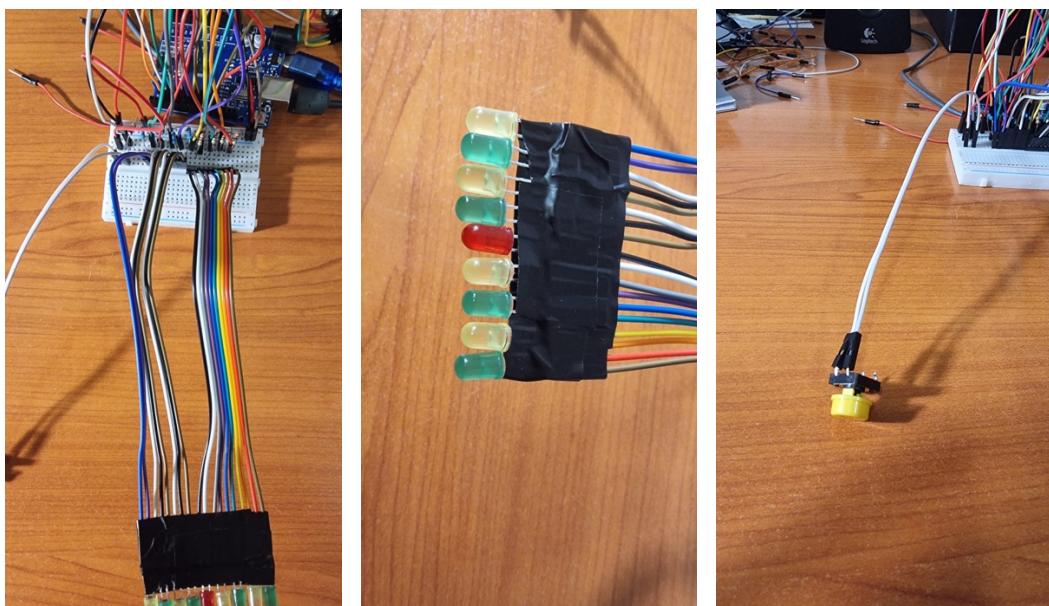
Na quarta parte do desenvolvimento do projeto fiz uma embalagem para meter as placas e todas as ligações.

Para a embalagem do jogo utilizei a caixa de um equipamento que tinha comprado há uns tempos – um disco externo.

As dimensões pareceram-me adequadas e tinha espaço generoso para acomodar todas as placas, peças e fios.



Tive de utilizar mais fios macho-fêmea para tirar os LEDs da placa e ligá-los na caixa de papelão. O mesmo tive de fazer para o botão de pressão e LCD. Para os LEDs, juntei-os todos, lado a lado, e prendi-os com fita isoladora preta.



Tirei as medidas de cada parte (LEDs, buzzer, LCD) que iria ficar visível no exterior da caixa e fiz os cortes necessários na caixa, recorrendo a um canivete suíço.

Para a entrada de energia:



Para o display (LCD):



Para o conjunto de LEDs:

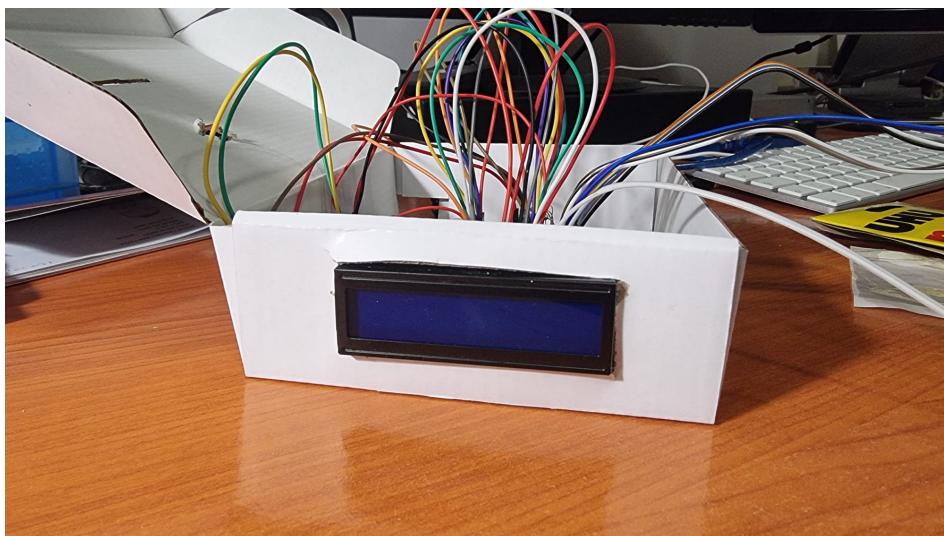
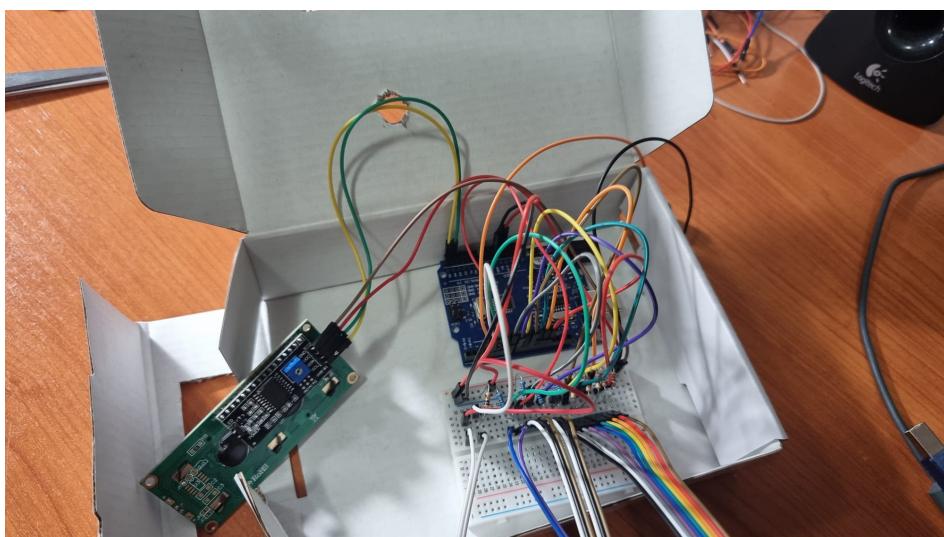


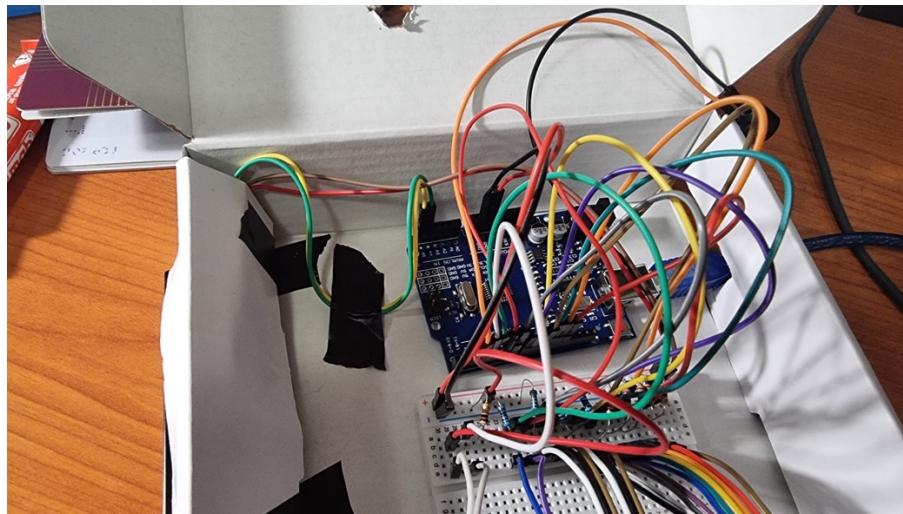
Para os LEDs e o buzzer:



Após os cortes da caixa comecei a acomodar e prender as peças e placas na caixa.

Acomodaçāo do LCD:

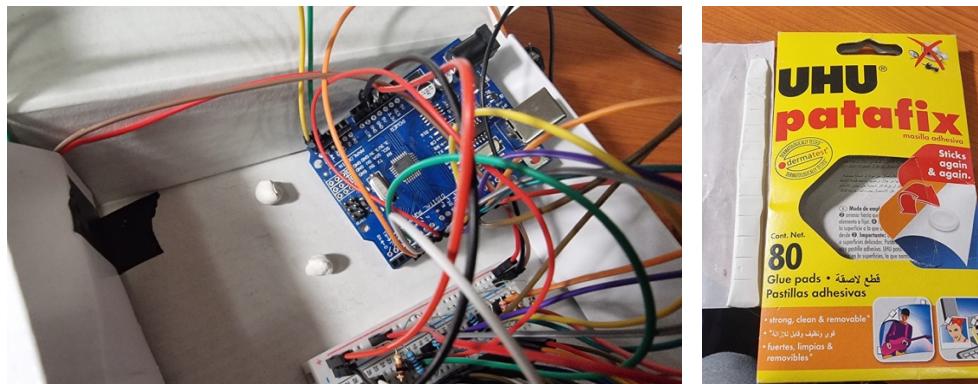




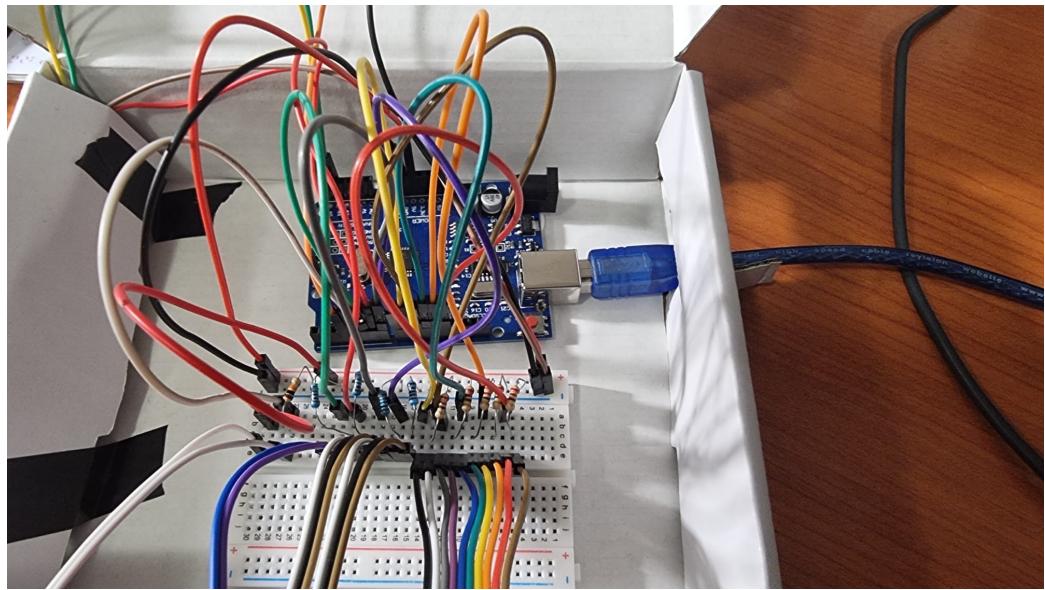
A acomodação da *breadboard* foi feita com a fita adesiva incorporada na mesma:



A acomodação da placa Arduino foi feita com pastilhas adesivas UHU:

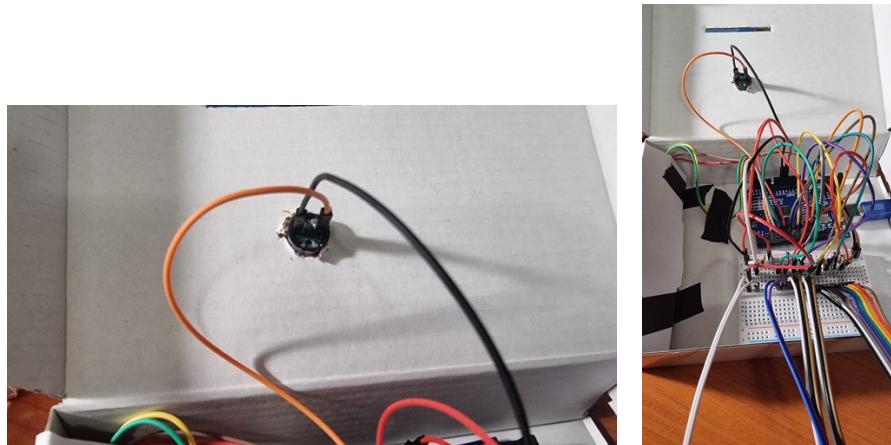


As ligações até aqui ficaram assim, com as placas bem presas:

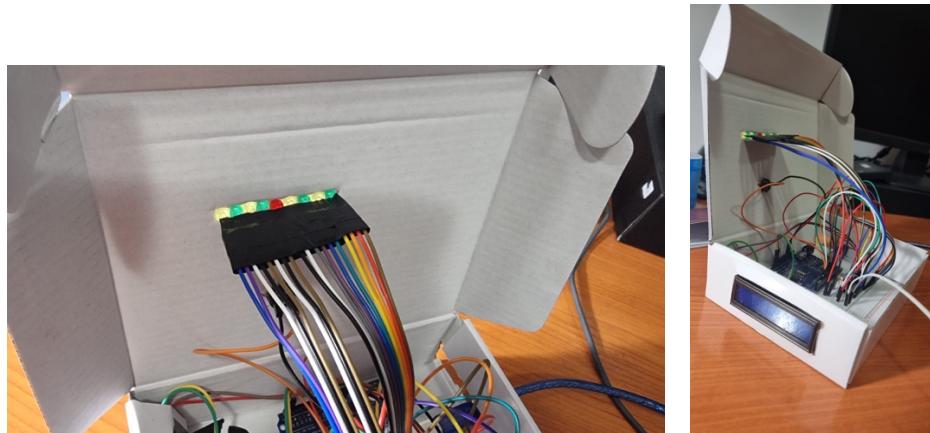


A partir daqui faltava colocar os LEDs e o buzzer na parte superior da caixa e fechar tudo.

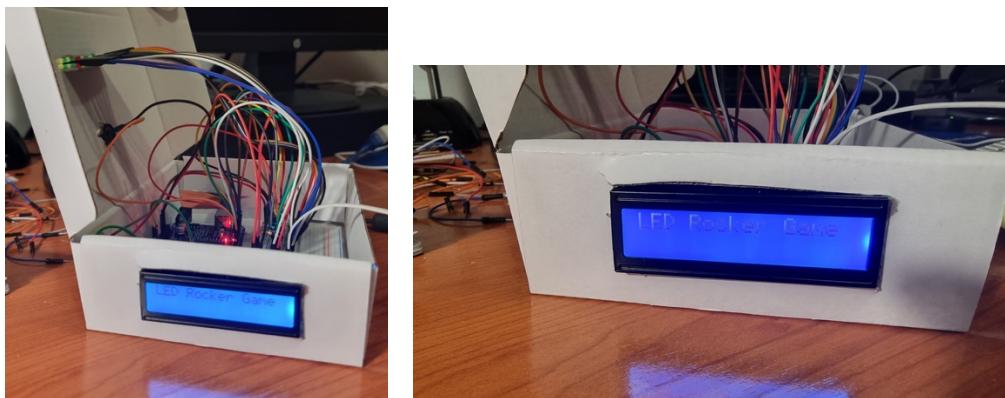
Acomodação do buzzer:



Acomodação dos LEDs:



Teste do funcionamento, após estas ligações:



Ligação do botão de pressão no exterior da caixa:



Esta ligação foi a mais difícil pois não tinha material para prender o botão em condições. Optei por passar os fios do botão pela ranhura do cabo de energia e liguei o botão na parte posterior da caixa, em que os dois pinos sem utilização ficaram presos na caixa, em cima e os outros dois não se viam (os fios ligaram aqui).

Finalização das ligações / caixa:





O resultado final da caixa é este, apresentado na imagem acima.

Resultados / Melhorias futuras

O trabalho ficou concluído com a elaboração da embalagem. O código foi inteiramente desenvolvido por mim. Consultei outros trabalhos semelhantes, mas utilizaram código / funções muito diferentes (específicas). Optei por desenvolver o meu código sem essas funções e ficou a funcionar.

Ao executar o jogo, após a conclusão dos trabalhos, verifiquei que o LED vermelho não estava a ligar. Concluí que o espaço interior da caixa, apesar de generoso, não foi suficiente para acomodar os fios da melhor forma, pois estes ocupam muito espaço interior e não são facilmente maleáveis.

Este é um ponto a melhorar, eventualmente, criar uma nova embalagem e ter mais cuidados na parte da ligação dos fios.

Outro aspeto que poderia ser melhorado é a ergonomia da caixa. A localização dos elementos na parte exterior não é a melhor. Se for realizada outra embalagem, os LEDs deviam ficar em paralelo com o LCD, por cima e o botão de pressão deveria ficar na parte da frente ou em cima, mas perto da frente, para facilitar o modo do jogo.

Apesar disso, a execução do jogo é na mesma possível e tudo o resto funciona.

No link seguinte pode ser visualizado um vídeo do funcionamento do jogo, com a embalagem completa:

https://drive.google.com/file/d/1jhnPVPj6ucwwAf5ngvGHKqQRlvQ11eSK/view?usp=share_link

https://drive.google.com/file/d/1UlxeobxjnVXoG8MDINGptgdvvWJg2ykz/view?usp=share_link

Vídeo Final do Projeto

Realizei um vídeo com todos os passos do desenvolvimento do projeto, que está guardado na pasta “Video Final”, dentro da pasta Projeto Final, no Google Drive (partilhado anteriormente, com todas as atividades realizadas). O vídeo pode ser acedido diretamente através do link:

https://drive.google.com/file/d/1H1wesuydMJWhyRzlrlWULSz5LS7_lTV/view?usp=share_link