



Matemática Furiosa

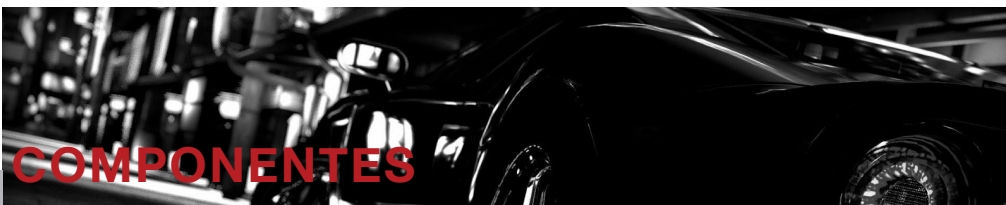
guia de montagem do interface físico

um jogo de André Filipe e Ricardo Pedro
um projeto Universidade Lusófona — ECATI (licenciatura em Videojogos)
& Casa Pia de Lisboa – CED Jacob Rodrigues Pereira
disponível em <<http://educacaoacessivel.ulusofona.pt>>

A BUSCA DE JACOB é um videojogo que pode ser jogado com um normal teclado
ou com um interface original concebido pelos autores do jogo.
Este documento contém as indicações fundamentais para a sua reprodução.



This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License



1 x Arduino Micro

https://www.ptrobotics.com/index.php?id_product=1776&controller=product

1 x Thumb Slide Joystick

https://www.ptrobotics.com/joysticks/1302-thumb-slide-joystick.html?search_query=joystick&results=45

1 x Push Button Momentary Black Panel Mount

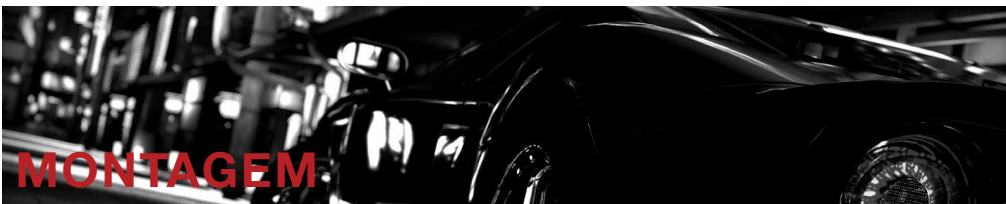
https://www.ptrobotics.com/push-button/2039-push-button-momentary-black-panel-mount.html?search_query=Button&results=370

1 x Comando Impresso em Impressora 3D

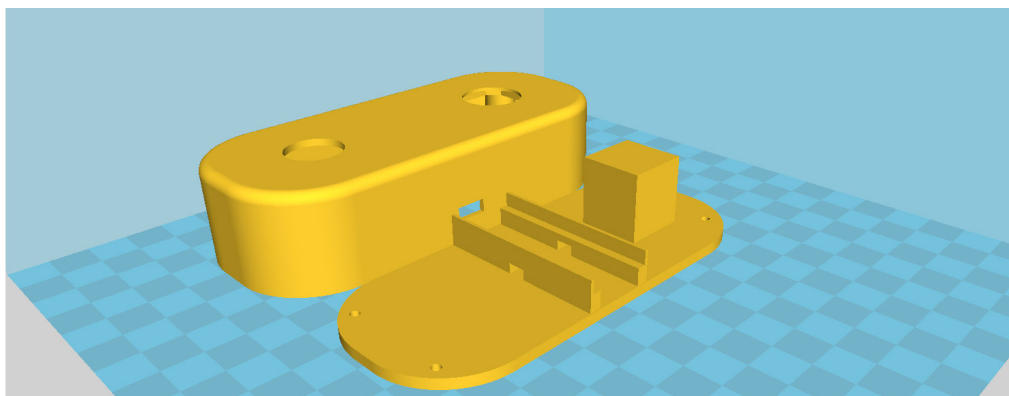
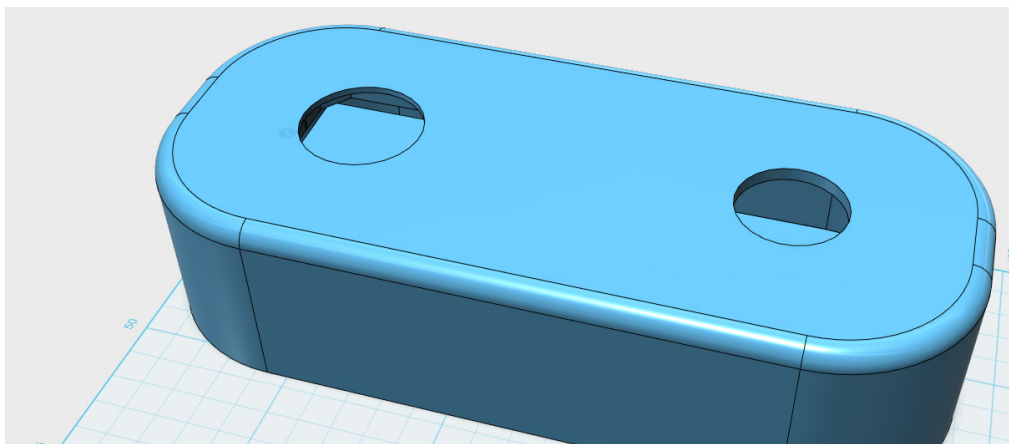
> nota: cortantes dos acrílicos (docs. illustator CS6 e PDF) e código arduino em ficheiros anexos.

Cortantes feitos a partir de < <http://www.makercase.com> >.



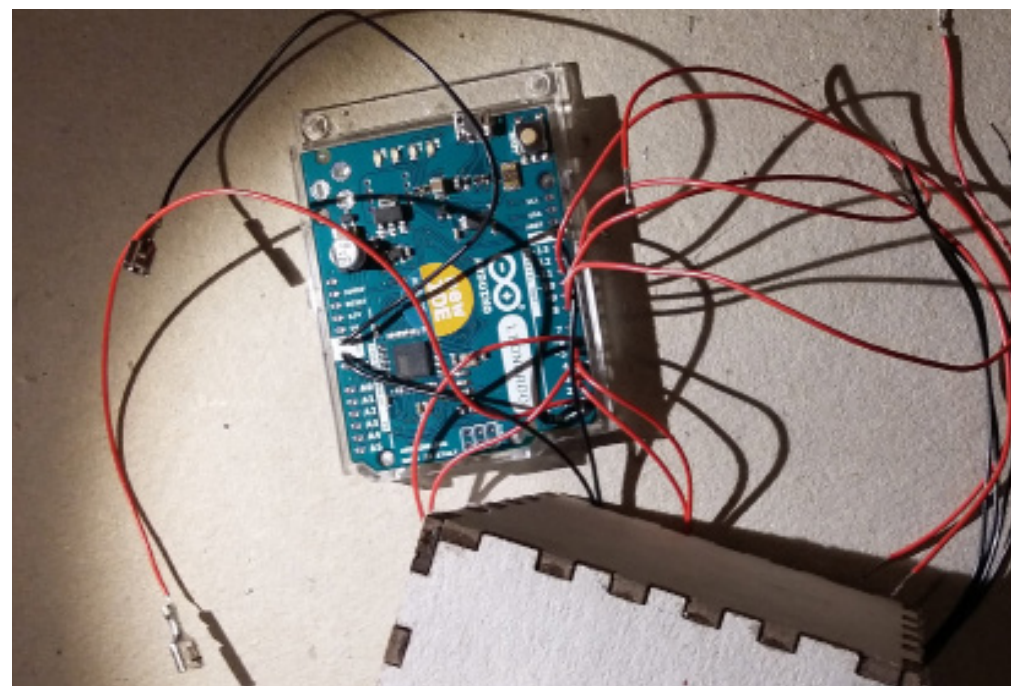


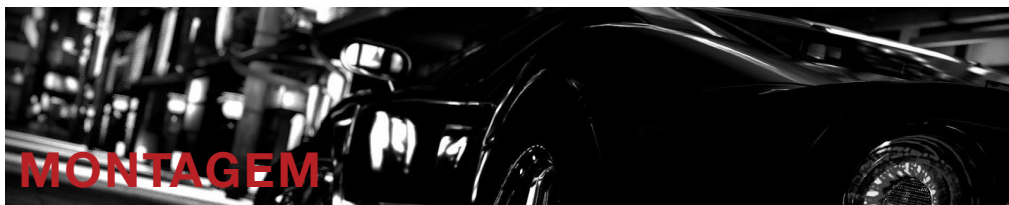
Em primeiro lugar, os comandos foram pensados e feitos utilizando o programa Adobe Illustrator e posteriormente gravados em SVG. Quando acabados, estes eram transferidos para o 123Design, onde iríamos construir um comando em 3D que seria usado para a impressão 3D do mesmo.



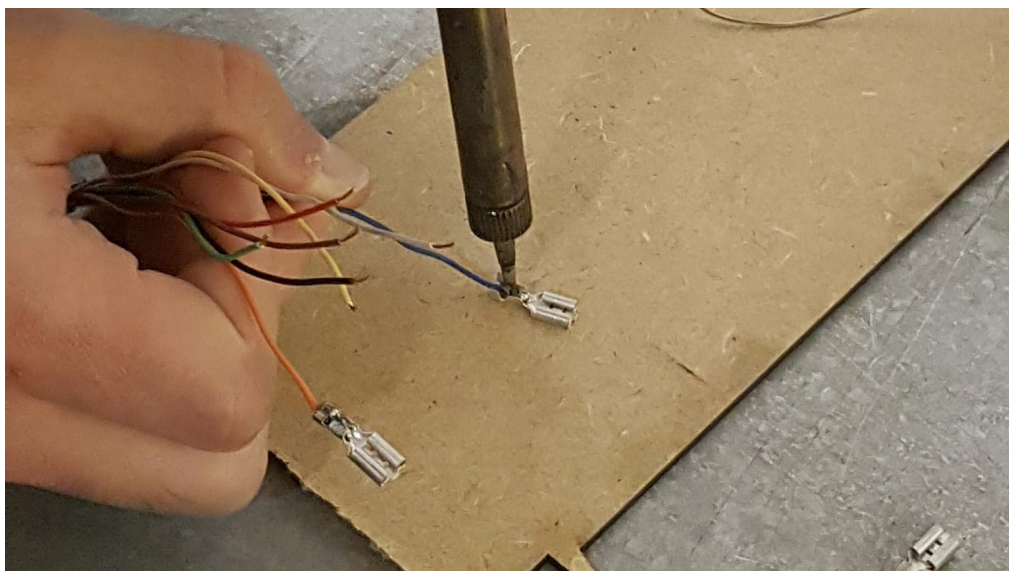
Após a impressão dos comandos fomos ajustando os moldes impressos de modo a encaixarem um no outro.

Com os comandos impressos e sem encontrarmos algum problema seguiu-se a solda dos componentes elétricos que mais tarde irão ligar o botão e o analógico ao Arduino.



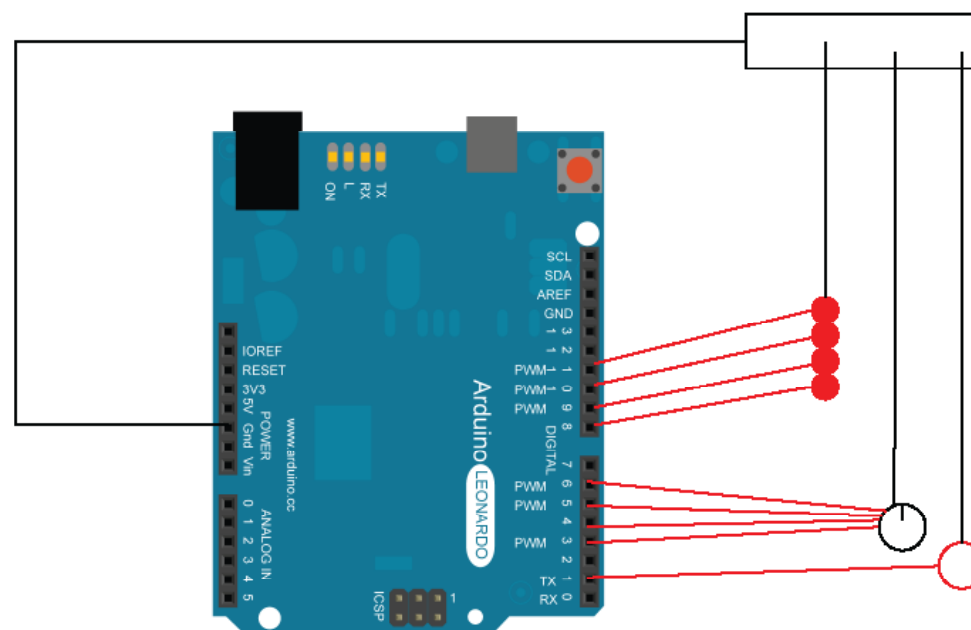


Primeiramente, os fios utilizados para as ligações são descarnados nas pontas de maneira a ficarem preparados para serem estanhados.



Depois dos componentes elétricos terem sido soldados foram ligados os analógicos às sockets analógicas do Arduino Micro, seguido de dois dos fios elétricos ao Ground e um dos fios do analógico foi ligado à saída de 9volts que se pode encontrar no Arduino.

O seguinte esquema pretende clarificar as ligações efetuadas apartir dos componentes do comando com o Arduino.





CÓDIGO

```
1 #include <Keyboard.h>
2
3 const int VERT = A1; // analog
4 const int HORIZ = A0; // analog
5 const int btn1 = 1;
6 const int btn2 = 0;
7
8 int btnReturn = 2;
9
10 char leftKey = KEY_LEFT_ARROW;
11 char rightKey = KEY_RIGHT_ARROW;
12 char upKey = KEY_UP_ARROW;
13 char downKey = KEY_DOWN_ARROW;
14
15 void setup() {
16     // put your setup code here, to run once:
17     pinMode(btnReturn, INPUT_PULLUP);
18
19     Serial.begin(9600);
20     Keyboard.begin();
21 }
22
23 void loop() {
24
25     if (digitalRead(btnReturn) == LOW) {
26
27         Keyboard.press(KEY_RETURN);
28
29         delay(10);
30
31         Keyboard.releaseAll();
32
33         delay(100);
34     }
35     // put your main code here, to run repeatedly:
36
37     int vertical, horizontal;
38
39     // read all values from the joystick
40
41     vertical = analogRead(VERT); // will be 0-1023
42     horizontal = analogRead(HORIZ); // will be 0-1023
43
44
45     Serial.print("X-axis: " + (String)horizontal);
46     Serial.flush();
47     delay(10);
48     Serial.println("Y-axis: " + (String)vertical);
49     Serial.flush();
50     delay(10);
51
52     if (digitalRead(btn1) == LOW) {
53         Keyboard.press((char) 0x20);
54         delay(100);
55     }
```

```
43
44
45     Serial.print("X-axis: " + (String)horizontal);
46     Serial.flush();
47     delay(10);
48     Serial.println("Y-axis: " + (String)vertical);
49     Serial.flush();
50     delay(10);
51
52     if (digitalRead(btn1) == LOW) {
53         Keyboard.press((char) 0x20);
54         delay(100);
55     }
56
57     else {
58         Keyboard.release((char) 0x20);
59         delay(10);
60     }
61
62     if(horizontal > 800) {
63         Keyboard.release(leftKey);
64         Keyboard.press(rightKey);
65         delay(10);
66     }
67
68
69     if(horizontal < 350) {
70         Keyboard.release(rightKey);
71         Keyboard.press(leftKey);
72         delay(10);
73     }
74
75
76     if(vertical > 700) {
77         Keyboard.release(downKey);
78         Keyboard.press(upKey);
79         delay(10);
80     }
81
82
83     if(vertical < 300) {
84         Keyboard.release(upKey);
85         Keyboard.press(downKey);
86         delay(10);
87     }
88
89     if(vertical > 300 && vertical < 700 && horizontal > 400 && horizontal < 750) {
90         Keyboard.releaseAll();
91         delay(10);
92     }
93
94
95 }
96
```



As imagens anteriores demonstram passo-a-passo a escrita do código

Primeiro começamos por chamar as portas onde ligamos previamente os componentes.

```
1 #include <Keyboard.h>
2
3 const int VERT = A1; // analog
4 const int HORIZ = A0; // analog
5 const int btn1 = 1;
6 const int btn2 = 0;
7
8 int btnReturn = 2;
```

Seguidamente foi aplicado a cada output do componente a um input de teclado, fazendo assim com que a cada movimento do analógico ou cada vez que o botão fosse pressionado fosse realizada a mesma ação que ao premir nas teclas do teclado.

Como por exemplo, o movimento do analógico corresponder aos inputs das setas direcionais.

```
7
8 int btnReturn = 2;
9
10 char leftKey = KEY_LEFT_ARROW;
11 char rightKey = KEY_RIGHT_ARROW;
12 char upKey = KEY_UP_ARROW;
13 char downKey = KEY_DOWN_ARROW;
14
```

Por fim, foi criado um if de maneira a que o analógico em posição de repouso não movimento o jogador, tal como aconteceria se o mesmo estivesse longe do teclado.

```
75
76 if(vertical > 700) {
77     Keyboard.release(downKey);
78     Keyboard.press(upKey);
79     delay(10);
80 }
81
82
83 if(vertical < 300) {
84     Keyboard.release(upKey);
85     Keyboard.press(downKey);
86     delay(10);
87 }
88
89 if(vertical > 300 && vertical < 700 && horizontal > 400 && horizontal < 750) {
90     Keyboard.releaseAll();
91     delay(10);
92 }
93
94
95 }
```