

PORTAL 2

Portal 2 guia para interface lúdico

um jogo de Diogo Sousa e Joana Ferreira

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

COMPONENTES

1 x Arduino Micro

https://www.ptrobotics.com/plataformaarduino-e-modelos-alternativos-equivalentes/3585-arduino-yun-mini.html?search_query=mini+arduino&results=137

1 x Placa Veroboard 65x95mm 24x36 buracos

<https://www.ptrobotics.com/pcb/412-veroboard-65x95mm-24x36buracos.html>

4 x Tactile Button 12mm Flat

<https://www.ptrobotics.com/tactile-switch/3267-tactile-button-12mm-flat.html>

2 x Tactile Button 12mm Red

<https://www.ptrobotics.com/tactile-switch/1048-tactile-button-12mm-red.html>

2 x Tactile Button 12mm Green

https://www.ptrobotics.com/tactile-switch/1047-tactile-button-12mm-green.html?search_query=tactile+button&results=39

2 x Tactile Button 12mm Black

https://www.ptrobotics.com/tactile-switch/1050-tactile-button-12mm-black.html?search_query=tactile+button&results=39

2 x KY-023 Dual-axis XY Joystick Module

https://www.ptrobotics.com/joysticks/5748-ky-023-dual-axis-xy-joystick-module.html?search_query=joystick&results=45

1 x Filamento Ultimaker PLA 2.85 750g Green

<https://www.ptrobotics.com/impressao-3d/4499-ultimaker-pla-285-750g-green.html>

1x Filamento Ultimaker PLA 2.85 750g Red

16x Crimp Connector Housing 1x1Pin

16 x Male Crimp Pin for 0.1" Housing

16 x PCB Header 40Pin Single Row

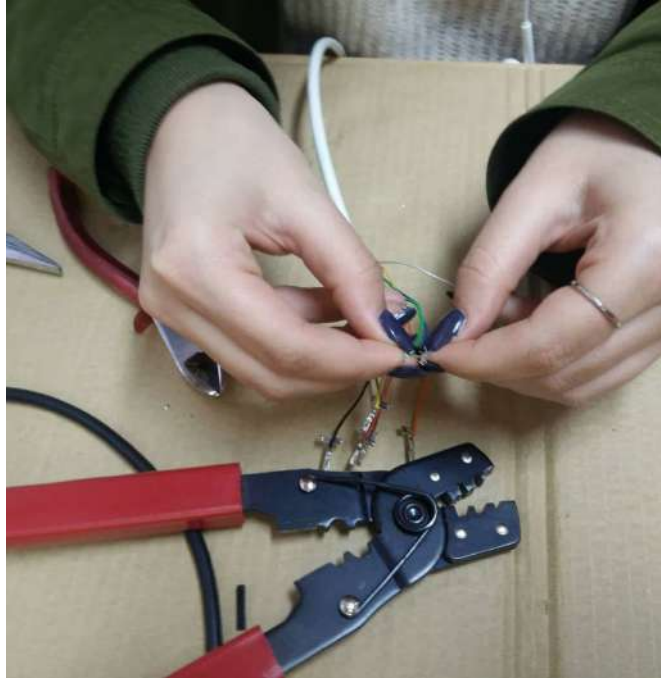
16 x PCB Header 40Pin Single Row

1 x Cola loctite

1 x Cabo blindado 8 condutores NCA8

Montagem

Primeiro descarnámos os fios que iríamos utilizar para que depois podessem ser soldados.



Depois deste processo a ligação foi feita da seguinte forma:

Comando Verde
E - Amarelo - Pin 0
Q - Vermelho - Pin 1
Espaço - Castanho - Pin 2
R - Azul - Pin 3
Eixo X - Verde - Pin A0
Eixo Y - Branco - Pin A1
5V - Laranja - Pin 5V

Comando Vermelho
O - Amarelo - Pin 4
U - Vermelho - Pin 5
Enter - Castanho - Pin 6
P - Azul - Pin 7
Eixo X - Verde - Pin A2
Eixo Y - Branco - Pin A3
5V - Laranja - Pin 3V

Depois do processo de montagem colocámos tudo nas posições certas no comando, aparafusámos o joystick à base do comando.

Após este processo fechámos os comandos com parafusos.



Programação Arduino

Primeiramente decidimos associar o arduino às teclas do teclado em vez de o ligar diretamente ao Unity.

Desta forma programámos da seguinte forma:

Declarámos a qual pin está associado a variável.

```
3 // PLAYER 1
4 //
5 const int player1ButtonA = 0;
6 const int player1ButtonB = 1;
7 const int player1ButtonC = 2;
8 const int player1ButtonD = 3;
9
10 const int player1XAxis = A0;
11 const int player1YAxis = A1;
12 int player1XValue = 0;
13 int player1YValue = 0;
14 //
15 // PLAYER 2
16 //
17 const int player2ButtonA = 4;
18 const int player2ButtonB = 5;
19 const int player2ButtonC = 6;
20 const int player2ButtonD = 7;
21 const int player2XAxis = A2;
22 const int player2YAxis = A3;
23 int player2XValue = 0;
24 int player2YValue = 0;
25
```

De seguida, registámos as variáveis como botões.

```
26 void setup()
27 {
28   pinMode(player1ButtonA, INPUT_PULLUP);
29   pinMode(player1ButtonB, INPUT_PULLUP);
30   pinMode(player1ButtonC, INPUT_PULLUP);
31   pinMode(player1ButtonD, INPUT_PULLUP);
32
33   //
34   // PLAYER 2
35   //
36   pinMode(player2ButtonA, INPUT_PULLUP);
37   pinMode(player2ButtonB, INPUT_PULLUP);
38   pinMode(player2ButtonC, INPUT_PULLUP);
39   pinMode(player2ButtonD, INPUT_PULLUP);
40   Keyboard.begin();
41 }
```

A linha 40, serve para começar a ler o teclado.

Quando um botão está primido fica em "Low<<

```
43 int treatValue(int data)
44 {
45   return (data * 9 / 1024) + 10 ;
46 }
```

Depois associamos cada botão do comando aos do teclado.

```
50 // PLAYER 1
51 //
52 if(digitalRead(player1ButtonA)==LOW )
53 {
54   Keyboard.write('E');
55   delay(500);
56 }
57 if(digitalRead(player1ButtonB)==LOW )
58 {
59   Keyboard.write('Q');
60   delay(500);
61 }
62 if(digitalRead(player1ButtonC)==LOW )
63 {
64   Keyboard.write(32);
65   delay(500);
66 }
67 if(digitalRead(player1ButtonD)==LOW )
68 {
69   Keyboard.write('R');
70   delay(500);
71 }
72
73 player1XValue = analogRead(player1XAxis);
74 delay(5);
75 player1YValue = analogRead(player1YAxis);
76 delay(5);
77
78 if(treatValue(player1XValue) > 16)
79 {
80   Keyboard.write('A');
81 }
82 else if(treatValue(player1XValue) < 12)
83 {
84   Keyboard.write('D');
85 }
86
87 if(treatValue(player1YValue) > 16)
88 {
89   Keyboard.write('W');
90 }
91 else if(treatValue(player1YValue) < 12)
92 {
93   Keyboard.write('S');
94 }
```

```
96 // PLAYER 2
97 //
98 if(digitalRead(player2ButtonA)==LOW )
99 {
100   Keyboard.write('O');
101   delay(500);
102 }
103 if(digitalRead(player2ButtonB)==LOW )
104 {
105   Keyboard.write('U');
106   delay(500);
107 }
108 if(digitalRead(player2ButtonC)==LOW )
109 {
110   Keyboard.write(176);
111   delay(500);
112 }
113 if(digitalRead(player2ButtonD)==LOW )
114 {
115   Keyboard.write('P');
116   delay(500);
117 }
118
119 player2XValue = analogRead(player2XAxis);
120 delay(5);
121 player2YValue = analogRead(player2YAxis);
122 delay(5);
123 if(treatValue(player2XValue) > 15)
124 {
125   Keyboard.write('J');
126 }
127 else if(treatValue(player2XValue) < 11)
128 {
129   Keyboard.write('L');
130 }
131
132 if(treatValue(player2YValue) > 14)
133 {
134   Keyboard.write('I');
135 }
136 else if(treatValue(player2YValue) < 11)
137 {
138   Keyboard.write('K');
139 }
140 }
141
```