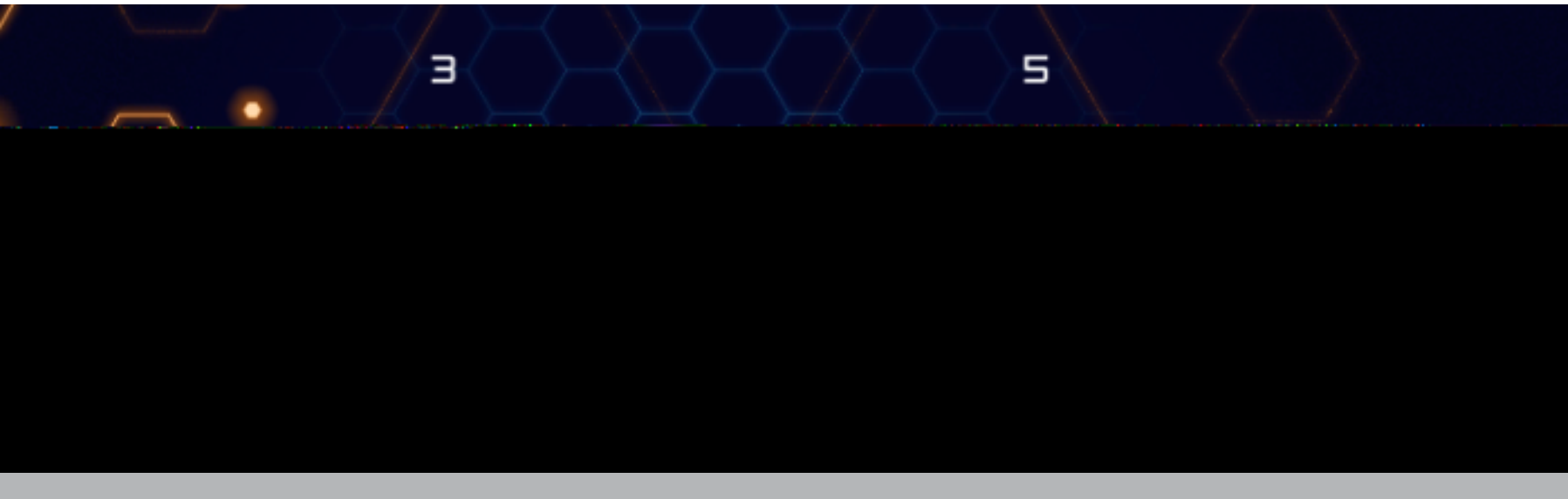




ReMat

guia para interface lúdico

um jogo de David Patrão, Gustavo Sobral e Miguel Gutierrez

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

ReMat 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

2 x Arduino Micro

<https://www.ptrobotics.com/plataformaarduino-e-modelos-alternativos-equivalentes/1033-arduino-uno.html>

1x Push-button red round

<https://www.ptrobotics.com/push-button/5942-push-button-domed-head-momentary-12mm-red.html>

1x Push-button blue round

<https://www.ptrobotics.com/push-button/5940-push-button-domed-head-momentary-12mm-blue.html>

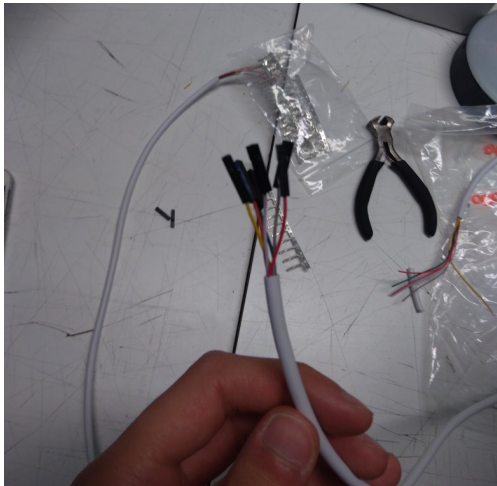
2x Thumb Joystick Module

https://www.ptrobotics.com/shields-varios/2845-itead-joy-stick-shield.html?search_query=joystick&results=54

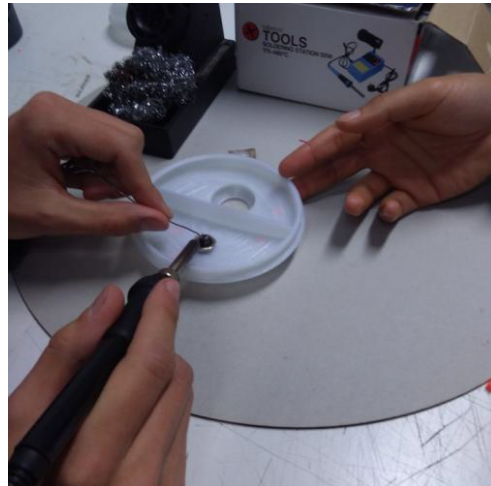
> nota: docs. illustrator CS6, 123d Design e PDF + código arduino em ficheiros anexos.



PROCESSO DE EXECUÇÃO



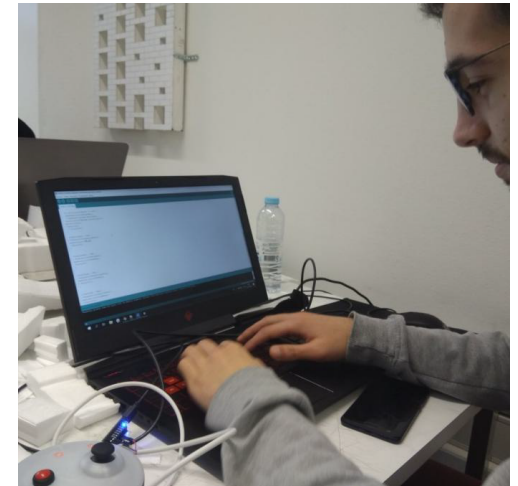
1º - Preparação de cabos para ligação ao Arduino.



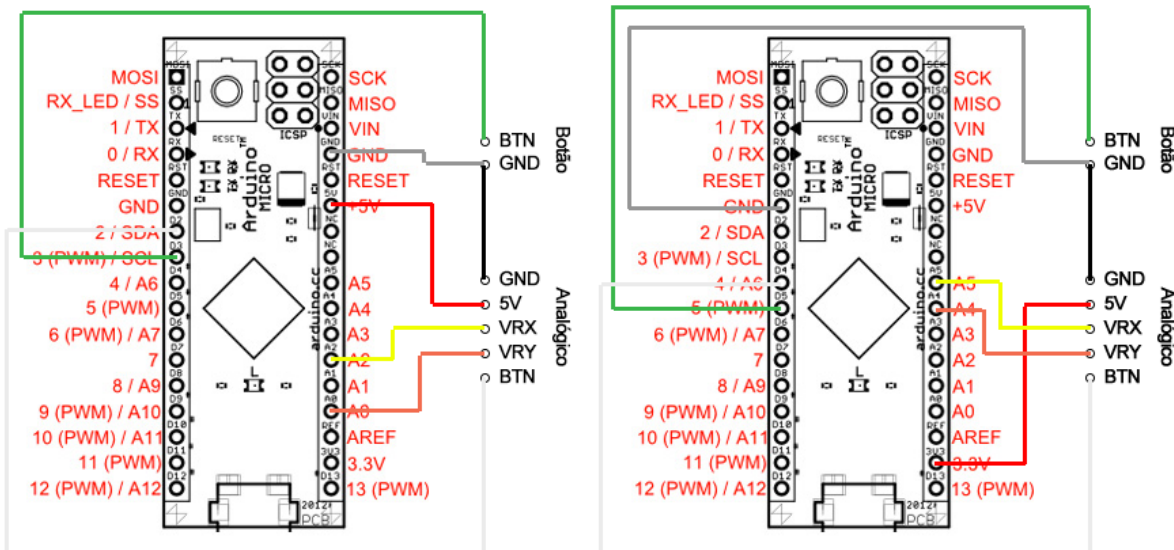
2º - Soldar a outra ponta dos cabos aos botões do comando.



3º - Ligação dos cabos anteriormente preparados ao Arduino.



4º - Programação dos controlos para o Arduino.



À esquerda temos o esquema de ligação do comando vermelho e à direita o do azul. Ambos os comandos partilham ligação com o mesmo Arduino, separámos em duas imagens para esta ser de mais fácil compreensão.

EXPLICAÇÃO DO CÓDIGO

Atribuição de teclas a cada botão de cada comando.

```
#include <Keyboard.h>

const int VERT = A0; // analog
const int HORIZ = A2; // analog
const int btn2 = 2; // sprint
const int btn3 = 3; // shoot
const int VERTblue = A4; // analog
const int HORTZblue = A5; //
const int btn4 = 4; // sprint
const int btn5 = 5; // shoot

char leftKey = KEY_LEFT_ARROW;
char rightKey = KEY_RIGHT_ARROW;
char upKey = KEY_UP_ARROW;
char downKey = KEY_DOWN_ARROW;
char sprintRed = 'M';
char shootRed = 'K';
char esquerdaKey = 'A';
char direitaKey = 'D';
char cimaKey = 'W';
char baixoKey = 'S';
char sprintBlue = 'Z';
char shootBlue = 32; // space

void setup(){
  //Put your setup code here, to run once:
  pinMode(btn2, INPUT_PULLUP);
  pinMode(btn3, INPUT_PULLUP);
  pinMode(btn4, INPUT_PULLUP);
  pinMode(btn5, INPUT_PULLUP);

  Serial.begin(9600);
  Keyboard.begin();
}
```

Botões quando estão LOW estão a ser pressionados.

```
if (digitalRead(btn2) == LOW) {
  Keyboard.press(sprintRed);
  Keyboard.release(sprintRed);
  Serial.print("Botão2 pressionado");
  Serial.flush();
}

if (digitalRead(btn4) == LOW) {
  Keyboard.press(sprintBlue);
  Keyboard.release(sprintBlue);
  Serial.print("Botão4 pressionado");
  Serial.flush();
}
```

Se o valor de horizontal for maior que 505, anda para a direita, se for menor que 490 anda para a esquerda

```
if(horizontal > 505) {
  Keyboard.release(downKey);
  delay(10);
  Keyboard.press(upKey);
  delay(10);
}
```

```
if(horizontalBlue > 401) {
  Keyboard.release(baixoKey);
  delay(10);
  Keyboard.press(cimaKey);
  delay(10);
}
```

```
if(horizontal < 490) {
  Keyboard.release(upKey);
  delay(10);
  Keyboard.press(downKey);
  //delay(10);
}
```

O mesmo que o horizontal se aplica para o vertical, mas em vez de direita e esquerda, é cima e baixo.

```
if(vertical > 505) {
  Keyboard.release(leftKey);
  delay(10);
  Keyboard.press(rightKey);
  //delay(10);
}
```

```
if(verticalBlue > 401) {
  Keyboard.release(esquerdaKey);
  delay(10);
  Keyboard.press(direitaKey);
  delay(10);
}
```

```
if(vertical < 490) {
  Keyboard.release(rightKey);
  delay(10);
  Keyboard.press(leftKey);
  //delay(10);
}
```

Se o vertical e horizontal estiverem entre estes valores, não exercem nenhuma função.

```
if(vertical > 490 && vertical < 505 && horizontal > 490 && horizontal < 505) {
  Keyboard.release(upKey);
  Keyboard.release(downKey);
  Keyboard.release(rightKey);
  Keyboard.release(leftKey);
  delay(10);
}
```