



Projeto 10

Controle Gráfico – Prática

Jan K. S. – janks@puc-rio.br

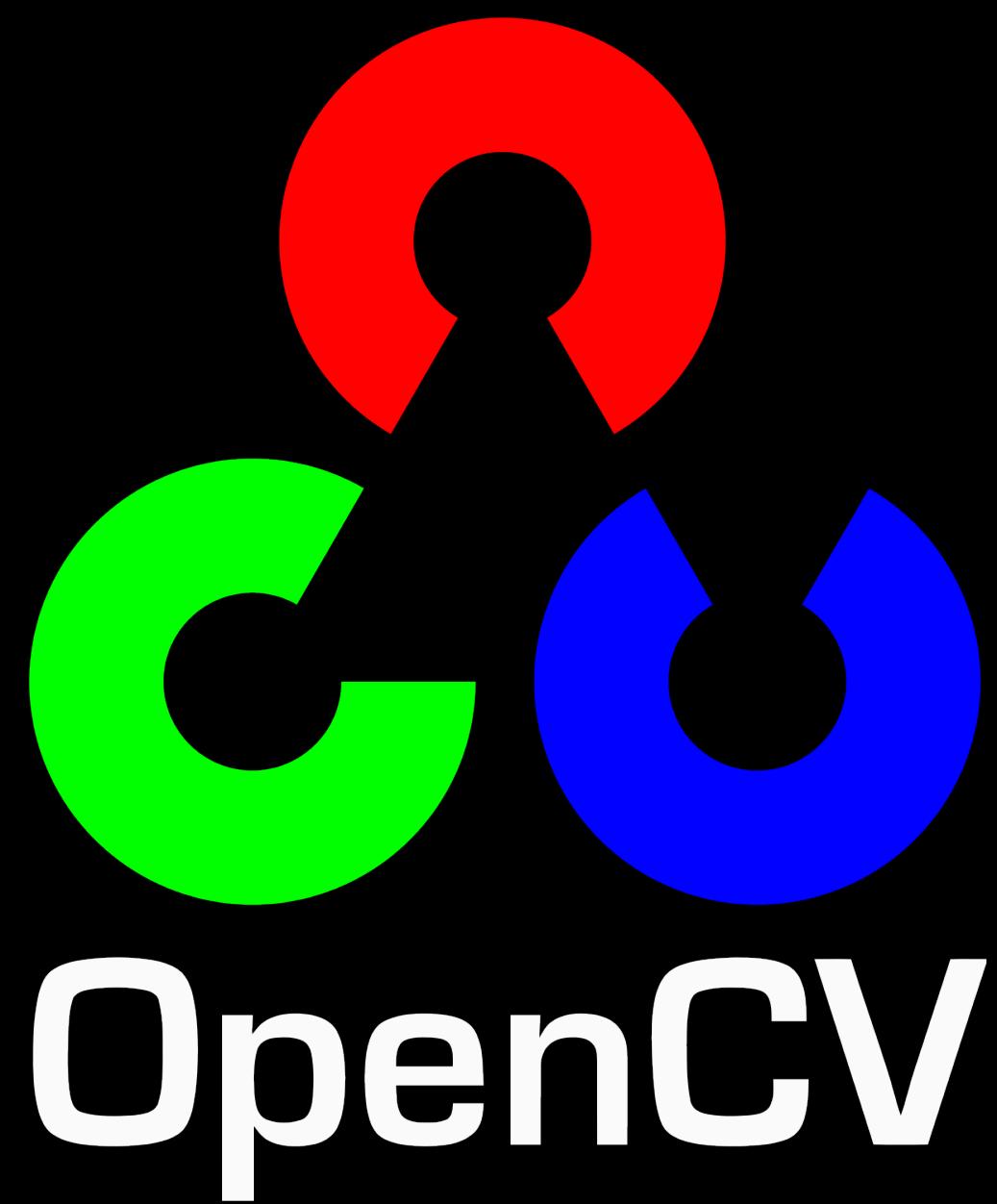
ENG1419 – Programação de Microcontroladores

Testes Iniciais

Testes Iniciais 1

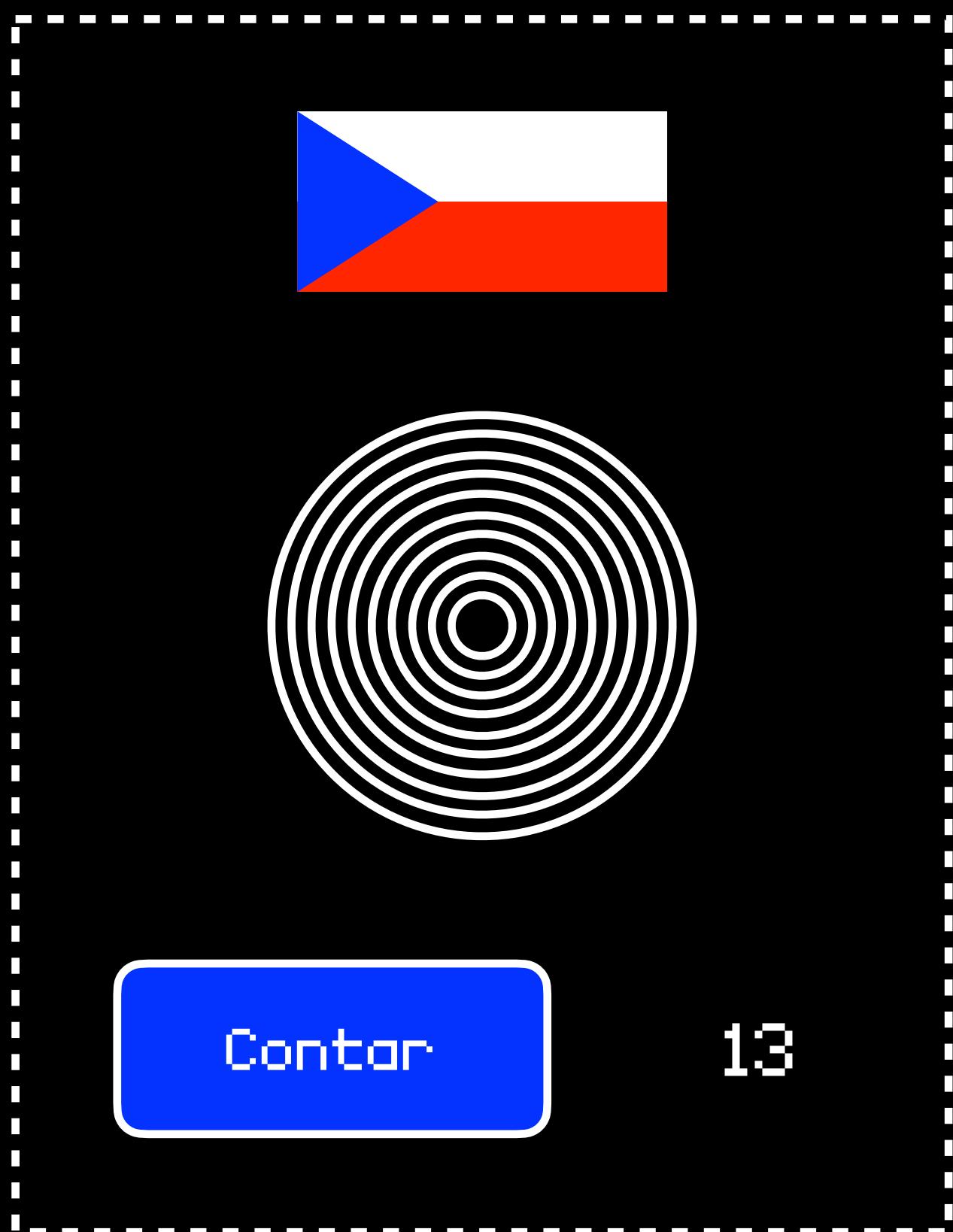


Testes Iniciais 2

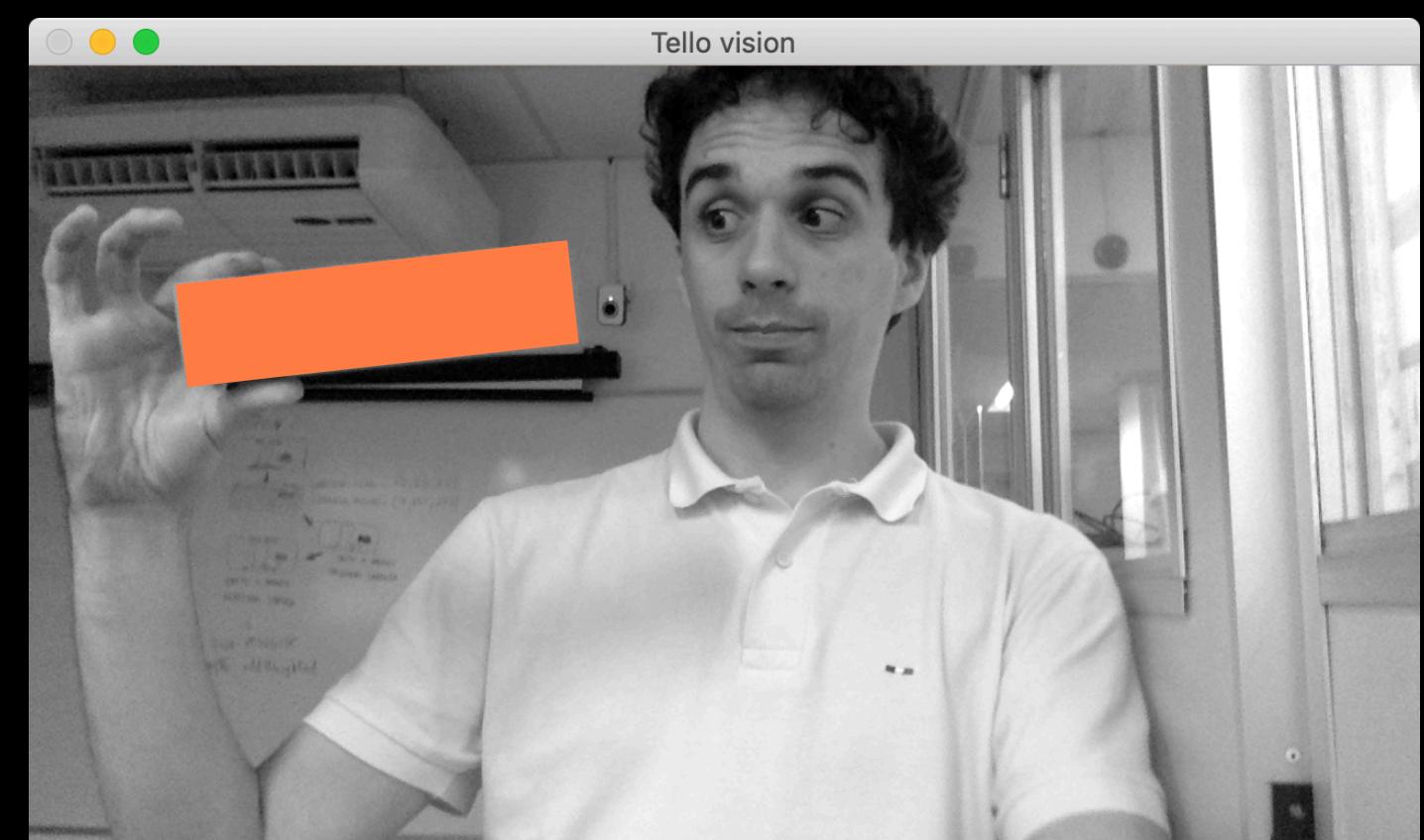


Duas Partes dos Testes Iniciais

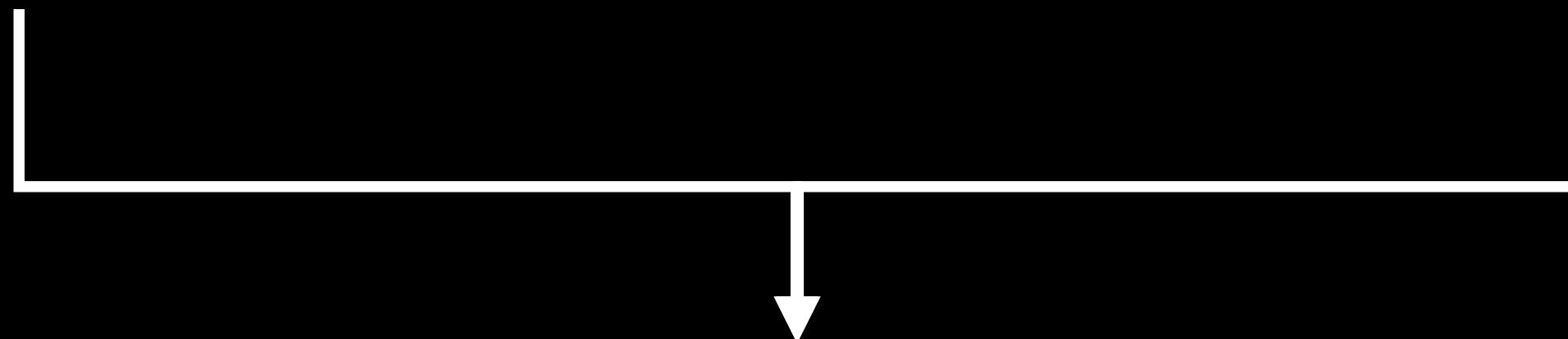
Testes Iniciais 1



Testes Iniciais 2



Duas Partes dos Testes Iniciais



Combinação de Duas Imagens para o Efeito Desejado



Testes Iniciais 01

Ao incrementar a contagem, **desenhe o número** ao lado do botão.
↳ DICA: lembre-se de apagar o texto anterior antes de desenhar o novo valor.

Desenhe a bandeira da República Checa no topo da tela.
Desenhe 10 círculos concêntricos de contorno branco no meio da tela, incrementando o raio de 5 em 5.
↳ DICA: use um for.

Desenhe um botão com o texto "Contar" no fundo da tela. Ao tocar no botão, incremente uma variável de contagem e imprima pela serial.



Testes Iniciais 02

Comece fazendo o **stream da câmera** direto na tela.

Crie e exiba uma imagem que mostre **apenas a cor laranja**, colocando o **resto como preto**.

↪ DICA: use os valores (0, 60, 60) e (20, 255, 255) para a faixa de cor HSV.

Crie e exiba uma outra imagem que mostre **substitua a cor laranja por preto** e que coloque o **resto em tons de cinza**.

↪ DICA: adapte o exemplo da máscara invertida da teoria.

Some as duas imagens anteriores, gerando uma que seja colorida nas regiões azuis e com tons de cinza no resto. Exiba essa imagem na tela.

↪ DICA: sim, é para usar o + mesmo.

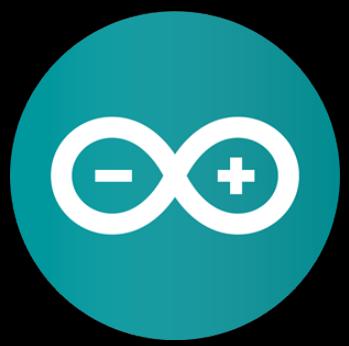
Escreva o nome do seu artista favorito no topo da imagem, com a **cor laranja**.

↪ DICA: lembre-se que a cor aqui está no formato BGR.

Implementação



Drone Tello



Controlador de Voo + Reconhecimento de Imagem

Implementação 01



Crie os botões na tela que enviem pela serial os textos "decolar" e "pousar" ao serem pressionados.

Crie os botões na tela que imprimam na serial os textos "esquerda", "frente" e "direita" ao serem pressionados e "parar" ao serem liberados.

↪ DICA: use a cor TFT_LIGHTGREY.

Desenhe um contorno retangular branco na parte inferior da tela, com dimensões 202x152. Acima dele, escreva "Objeto Laranja Detectado".



Implementação 02

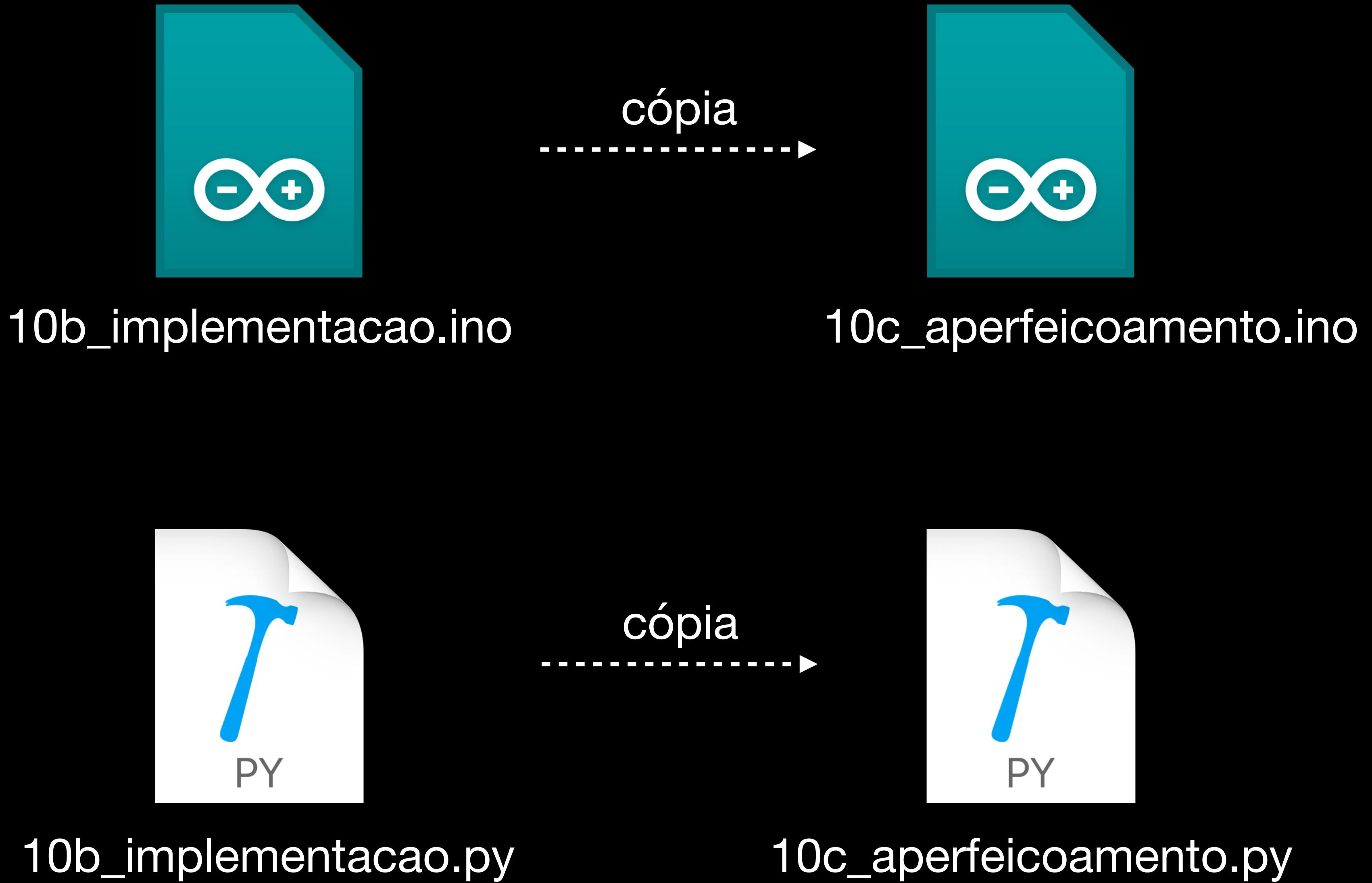
Detecte regiões com a cor laranja dos Testes Iniciais na imagem da câmera, e desenhe retângulos verdes ao redor delas.

↪ DICA: comece usando o exemplo de Detecção de Contornos e Retângulos da teoria.

Modifique o código acima para desenhar apenas o retângulo que tiver a maior área, e apenas se essa área for maior que 2000.

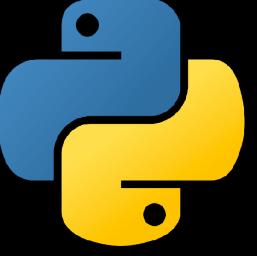
↪ DICA: crie variáveis antes do for para armazenar as coordenadas e dimensões do retângulo maior.

Aperfeiçoamento



Cópia do Código da Implementação para o Aperfeiçoamento



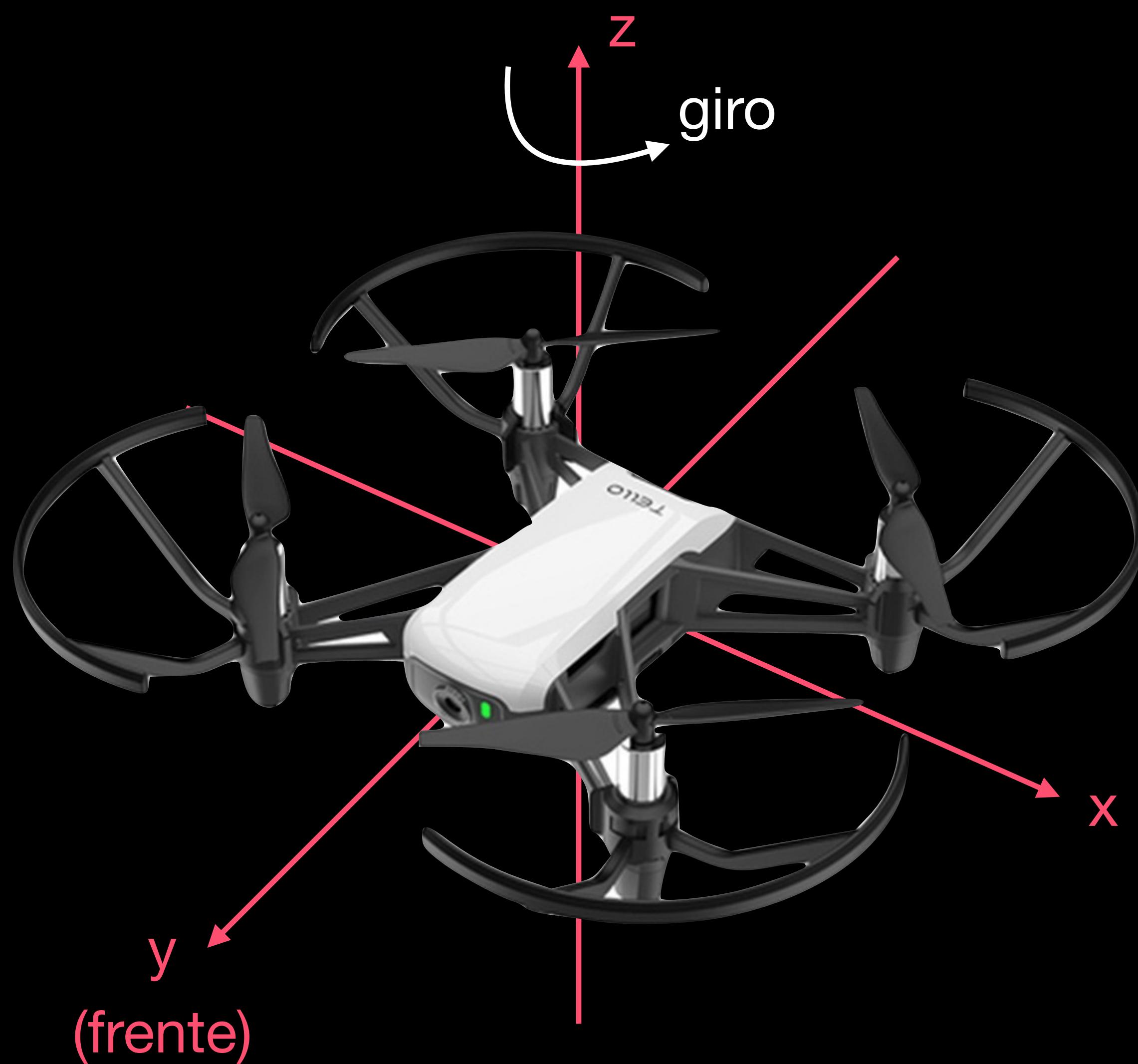
 python



"decolar"
"pousar"
"esquerda"
"frente"
"direita"
"parar"

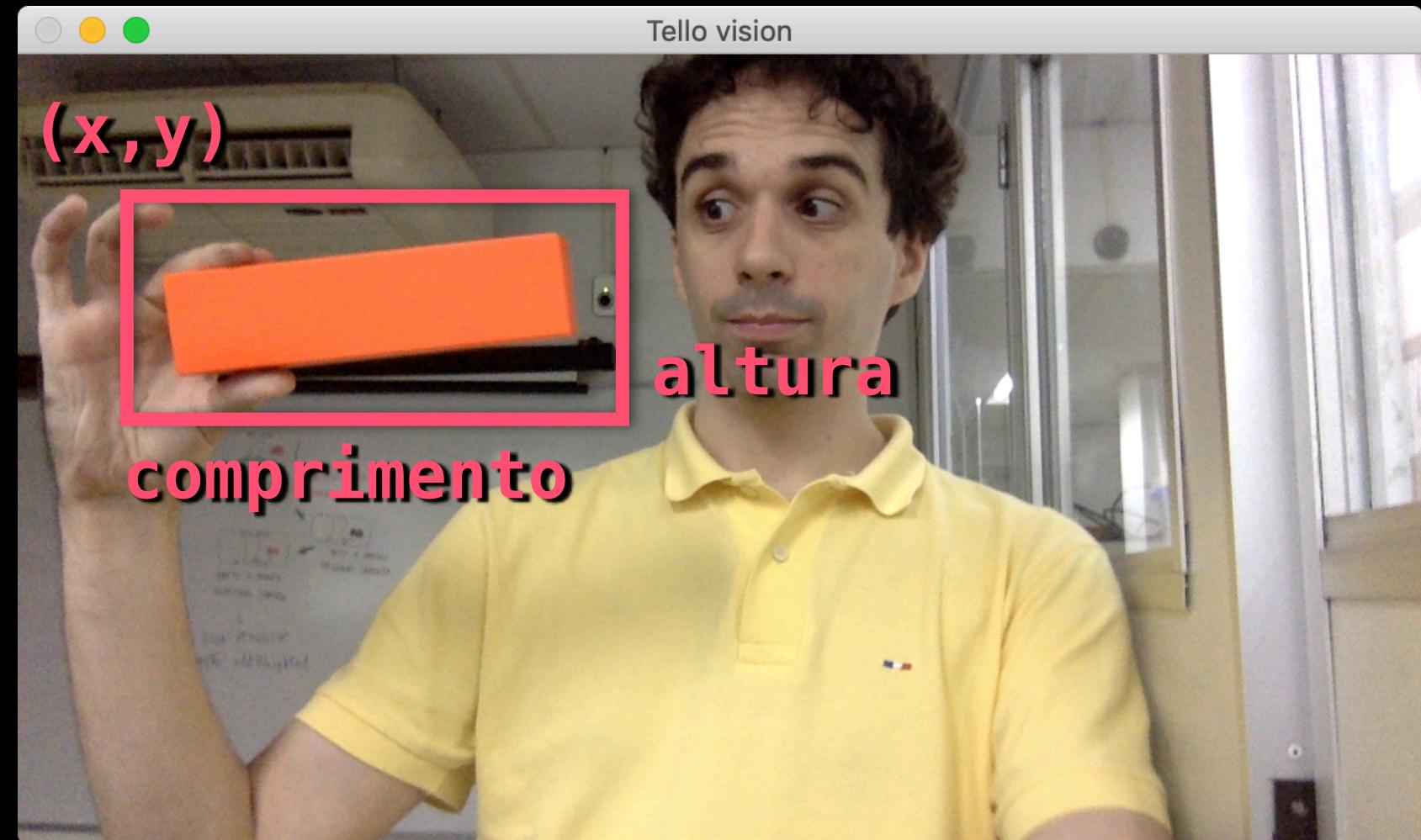


Controlador de Voo + Reconhecimento de Imagem

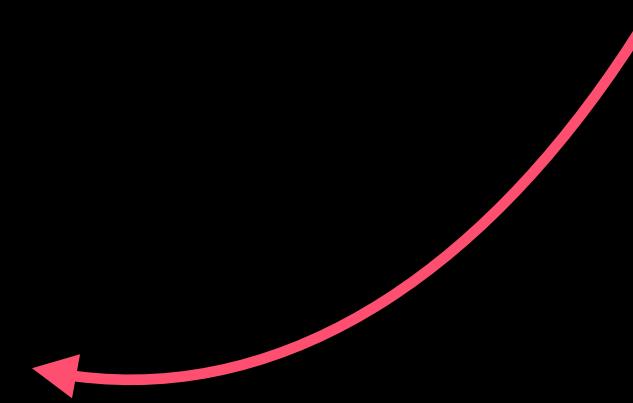


Coordenadas do Drone Tello

```
>>> from extra.tello import Tello  
  
>>> drone = Tello("TELLO-D023AE", test_mode=True)  
>>> drone.inicia_cmds()  
  
>>> drone.takeoff()  
>>> drone.land()  
>>> drone.goto(deltaX, deltaY, deltaZ, velocidade)  
>>> drone.rc(velocidade_x, velocidade_y, velocidade_z,  
velocidade_giro) # velocidades entre -100 e 100  
>>> imagem = drone.current_image  
>>> drone.state # dados como bateria, wifi, etc  
{'mid': 32.0, 'x': 0.0, 'y': 0.0, 'z': 0.0, 'mpry':  
'0,0,0', 'pitch': 1.0, 'roll': 0.0, 'yaw': 53.0,  
'vgx': 0.0, 'vgy': 0.0, 'vgz': 0.0, 'templ': 82.0,  
'temph': 83.0, 'tof': 10.0, 'h': 0.0, 'bat': 90.0,  
'baro': -111.55, 'time': 4.0, 'agx': 21.0, 'agy':  
-9.0, 'agz': -1000.0}
```



alturaImagem = ?



x e comprimento → multiplicar por comprimentoTela / comprimentoImagem
y e altura → multiplicar por alturaTela / alturaimagem

Mapeamento das Coordenadas e Dimensões para a Tela do Arduino



Aperfeiçoamento 01

Ao receber um texto na Serial no formato "retangulo 020 030 100 032", imprima o texto recebido e **desenhe um retângulo laranja** na posição e dimensões especificadas dentro da área. Envie alguns comandos pela serial para testar.

↪ DICA: use a `.substring` para pegar os valores. É necessário somar um valor de deslocamento nas coordenadas recebidas, para desenhar dentro do contorno retangular do item anterior.

Apague o retângulo laranja anterior sempre que receber um novo comando da Serial.

Inicie a Serial1. Ao clicar nos botões virtuais, **envie o texto pela Serial1 também**. E, no código do item anterior, **mude a leitura para a Serial1**.

Teste a conexão serial entre o seu Arduino e o Python da outra dupla.



Aperfeiçoamento 02

Ao receber os textos "decolar", "pousar", "esquerda", "frente", "direita" e "parar" pela serial, envie comandos para o drone (gire-o no caso da "esquerda" ou "direita"). Teste esse recebimento usando o Arduino de testes.
↪ DICA: use os comandos `drone.takeoff`, `drone.land` e `drone.rc`. Use -40, 0 ou 40 como valores de velocidade.

Salve as dimensões da imagem numa variável global.
↪ DICA: pesquise no Google como obter o comprimento e altura de imagens no OpenCV.

Use o timer recorrente de 2 segundos para imprimir no Shell as coordenadas e as dimensões do retângulo detectado, no formato "retangulo 020 030 100 032" (3 dígitos para cada número). Esses valores devem estar mapeados para as dimensões 200 x 150 do contorno retangular branco do Arduino.
↪ DICA: faça a regra de três que eu expliquei uns slides atrás.

Após imprimir no shell, envie essa mesma string pela Serial. Teste esse envio usando o Arduino de testes.

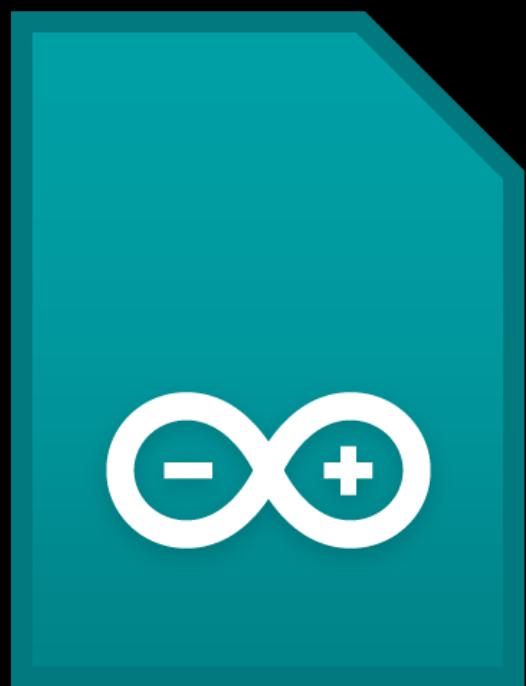
Teste a conexão serial entre o Arduino da outra dupla e o seu Python.

Desafio Extra



10c_aperfeicoamento.ino

cópia



10d_desafio.ino



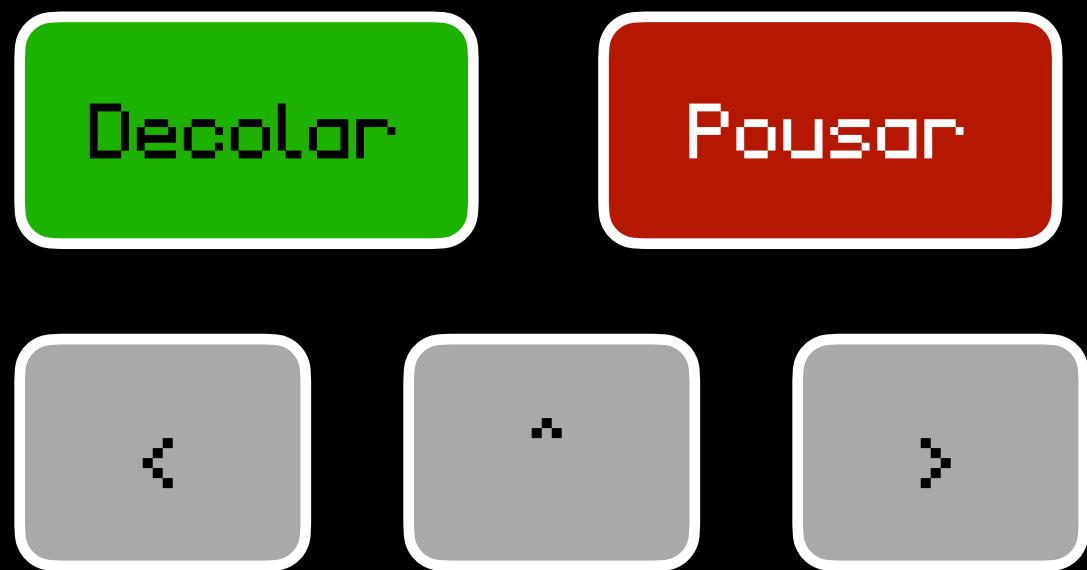
10c_aperfeicoamento.py

cópia

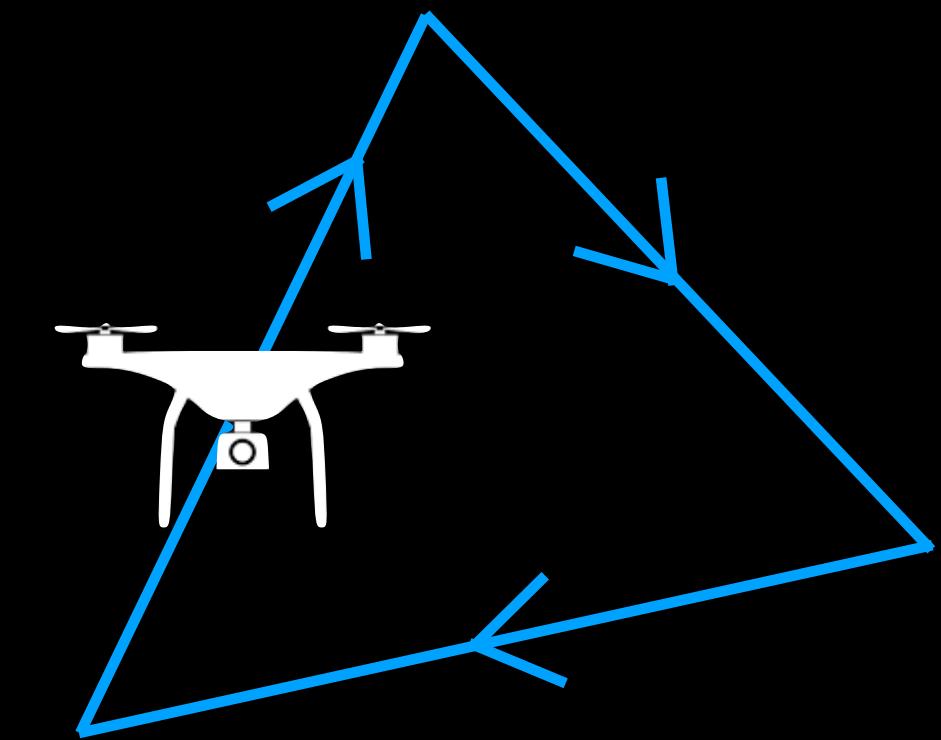
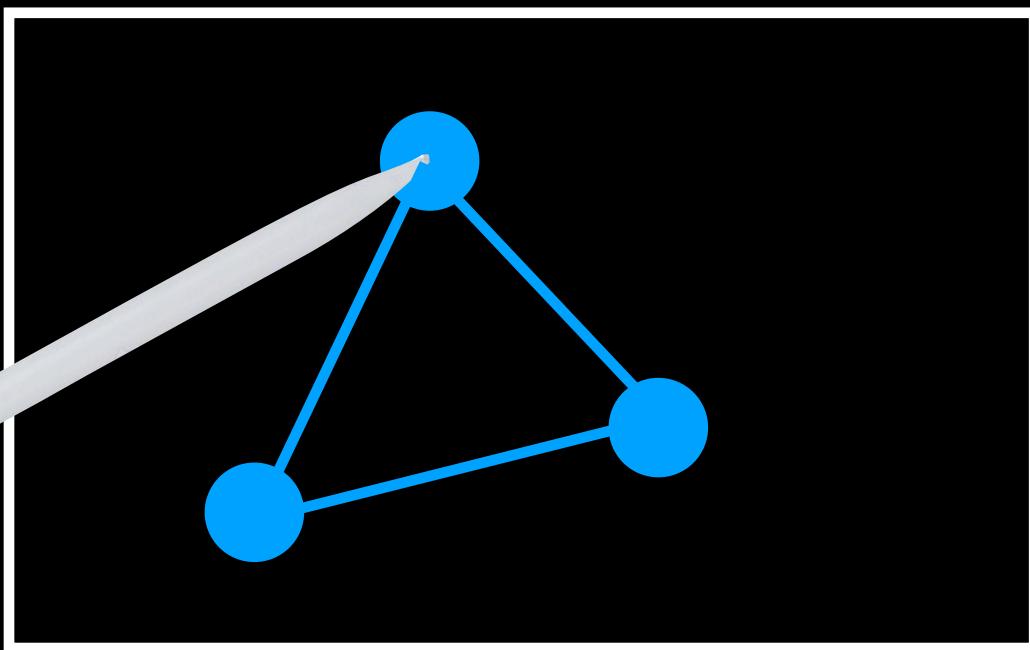


10d_desafio.py

Cópia do Código do Aperfeiçoamento para o Desafio



100 pixels = 50 cm



"trajeto 14 50 30 100 70 39 ..."

x1 y1 x2 y2 x3 y3

Trajeto de Voo Personalizado



Desafio 01

Ao tocar **dentro da área inferior**, marque um círculo azul na **posição**.

Ao tocar novamente na área, **marque outro ponto e trace uma reta dele** até o ponto anterior.

Armazene as coordenadas x e y de cada ponto coletado. Faça um ajuste nas coordenadas para que o x fique entre 0 e 200 e o y entre 0 e 150.

↪ **DICA:** você pode usar a biblioteca LinkedList ou um vetor bem grande.

Ao tocar bem próximo do primeiro ponto, **feche o polígono e envie os pontos coletados pela serial** no formato o texto "trajeto 14 50 30 100 70 39 ...". **Não inclua o ponto que fechou o polígono**, porque ele é redundante com o primeiro ponto.

Ao marcar o primeiro ponto, **pare de desenhar o retângulo** ao receber o comando pela serial. Ao fechar o polígono, **volte a desenhar esse retângulo** ao receber o comando.

Teste a conexão serial entre o seu Arduino e o Python da outra dupla.



Desafio 02

Ao receber um texto pela serial no formato "trajeto 14 50 30 100 70 39 ...", crie duas listas com as variações de coordenadas x e y, mapeando 100 pixels para 50 cm.
↪ DICA: use a função split.

Em seguida, crie um novo timer recorrente de 5 segundos que mande o drone ir para o próximo ponto do trajeto, circulando para sempre.
↪ DICA: use o comando drone.goto.

Ao receber um dos comandos da Implementação, cancele o timer acima.

Teste a conexão serial entre o Arduino da outra dupla e o seu Python.



janks.link/micro/projeto10.zip

Material do Projeto 10