

Tópico	IA DE RECONHECIMENTO DE GESTOS		
Descrição da Aula	Passando do uso de modelos de rede neural pré-construídos com métodos padrão para o uso de redes neurais para controlar objetos na tela usando gestos manuais, o aluno vai aprender a estender a funcionalidade dos modelos de aprendizado de máquina e moldá-la para se adequar às suas soluções de aplicativos da web.		
Aula	ADV-C125		
Duração da Aula	55 min		
Objetivos	<ul> <li>Buscar as coordenadas x e y do nariz</li> <li>Desenhar um quadrado usando as coordenadas x e y do nariz</li> <li>Buscar a coordenada x do pulso esquerdo e direito</li> <li>Descobrir a diferença entre a coordenada x do pulso esquerdo e direito</li> <li>Remover os decimais usando a função floor()</li> <li>Atualizar o tamanho do quadrado</li> </ul>		
Recursos Necessários	<ul> <li>Recursos da Professora         <ul> <li>Utilize suas credenciais de login do gmail</li> <li>Fone de ouvido com microfone</li> <li>Webcam</li> <li>Bloco de notas e caneta</li> </ul> </li> <li>Recursos do Aluno         <ul> <li>Utilize suas credenciais de login do gmail</li> <li>Fone de ouvido com microfone (opcional)</li> <li>Webcam</li> <li>Bloco de notas e caneta</li> </ul> </li> </ul>		
Estrutura da Aula	Introdução Atividade Dirigida pela Professora Atividade Dirigida pelo Aluno Encerramento Dicas e Sugestões do Projeto	5 min 10 min 30 min 5 min 5 min	

Observação: Certifique-se de que o aluno habilitou o webGL para o navegador, para melhorar a performance do aplicativo.

Se o aplicativo travar no Chrome, tente usar o Mozilla Firefox (com o webGL habilitado). As instruções para webGL são fornecidas na <u>Atividade da Professora - 5</u> e na <u>Atividade do Aluno - 2</u>.



## SESSÃO DE INTRODUÇÃO - 5 min

A professora inicia a apresentação de slides

do slide 1 ao 16.

Consulte as notas da oradora e siga as instruções em cada slide.

ATIVIDADE DA PROFESSORA



- 10 min

Peça ao aluno para entrar em modo tela cheia.

## A PROFESSORA INICIA O COMPARTILHAMENTO DE TELA

Dizer Fazer

Por favor, siga o fluxo da aula:

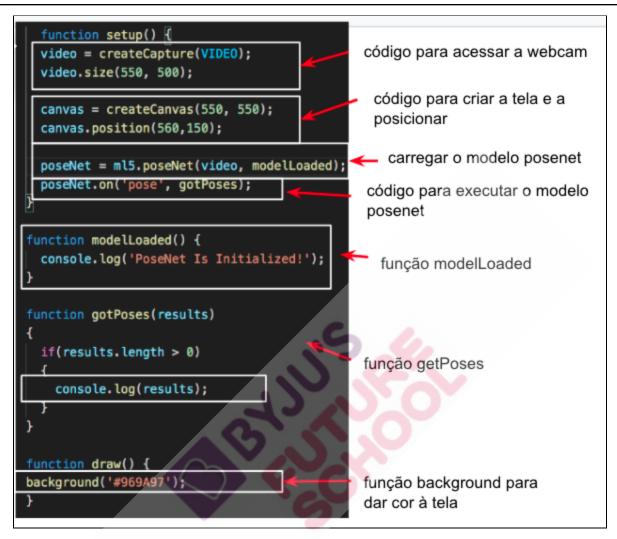
Explique o código JS.

Deixe o aluno fazer o código JS.

Depois, se tiver tempo, faça as Atividades Adicionais.

Na última aula, iniciamos o código JS para o aplicativo web. E concluímos essa parte do código:





Portanto, continuaremos programando o código JS no arquivo main.js para completar o app hoje mesmo.

# Nossa primeira tarefa será buscar as coordenadas x e y do nariz e, usando essas coordenadas, mover o quadrado na tela.

1. Primeiro vamos definir 2 variáveis para manter as coordenadas x e y do nariz e inicializar com um valor de 0, definir essas variáveis no início do arquivo **main.js**:

```
noseX=0;
noseY=0;
```



Na última aula, na função gotPoses obtivemos o result (coordenadas x e y de 17 partes do corpo) e havíamos consolado, certo?

```
function gotPoses(results)
 if(results.length > 0)
   console.log(results);
```

Execute a **Atividade da Professora - 1** e abra a tela do console.

Saída na tela do console:

```
PoseNet Is Initialized
                                                      main.js:17
                                                     main.js:24
▶ [{...}]
```

Precisamos ler este objeto e buscar as coordenadas x e y do nariz, enquanto lemos o objeto também escreveremos o código:

1. Primeiro clique na seta para expandir:



Queremos ler os objetos de **resultados**, então primeiro vamos escrever



2. Em seguida, clique na seta ao lado de [ para expandir:

```
main.js:24
0: {pose: {...}, skeleton: Array(0)}
length: 1
 _proto__: Array(0)
```

Clicamos no índice 0 que está no objeto de resultados para que o código seja:



## results[0]

Em seguida, clique na seta ao lado de pose: para expandir:

```
main.js:24

* [{--}] 

pose: {score: 0.25857010390866303, keypoints: Array(1...

| skeleton: []

| __proto__: Object
| length: 1

| __proto__: Array(0)
```

• Então, no índice 0, clicamos no objeto de pose, então o código será:

## results[0].pose

Essa quantidade de código permanecerá a mesma sempre que quisermos acessar qualquer parte do corpo.

- Então no objeto pose há duas partes importantes <u>keypoints</u>: e 17 partes do corpo com coordenadas x e y.
- o <u>keypoints</u>: tem a mesma coisa, as 17 partes do corpo com coordenadas x e y.
- Portanto, não precisamos entrar em <u>keypoints</u>, porque também encontramos todas as coordenadas x e y das 17 partes do corpo fora de keypoints:

OBS: Professora mostre ao aluno onde está <u>keypoints</u> e as demais partes do corpo, como mostrado na imagem



```
▼ [{...}] []
 -0:
     ▶ keypoints: (17) [{_}, {_}, {_}, {_}, {_}, {_}],
     ▶ leftAnkle: {x: 216.4462605415032, y: 301.3378774910.
     ▶ leftEar: {x: 188.69164561667637, y: 161.36775602374...
     ▶ leftElbow: {x: 268.9376267773366, y: 303.8666393324.
     ▶ leftEye: {x: 162.2530000251636, y: 154.137602465891.
     ▶ leftHip: {x: 213.40800711983127, y: 297.16762311277...
     ▶ leftKnee: {x: 209.40526075530468, y: 297.9268546690...
     ▶ leftShoulder: {x: 223.48277129625018, y: 258.861313.
     ▶ leftWrist: {x: 263.52472737518667, y: 303.070325461.
     ▶ nose: {x: 145.55414210983187, y: 178.28064153068942.
     ▶ rightAnkle: {x: 103.52699115262394, y: 299.35830692.
     ▶ rightEar: {x: 114.41234552372268, y: 175.0747362772...
     ▶ rightElbow: {x: 49.633986489814625, y: 308.25721171
     ▶ rightEye: {x: 130.25980309436196, y: 159.9434983660.
     ▶ rightHip: {x: 102.40974590095162, y: 301.0057866224...
     ▶ rightKnee: {x: 100.08567090620073, y: 299.290570259...
     ▶ rightShoulder: {x: 83.36848592200474, y: 252.199065...
     ▶ rightWrist: {x: 42.54400660420021, y: 304.860674428.
      score: 0.30651417103431683
     ▶ __proto__: Object
   ▶ skeleton: []
   ▶ __proto__: Object
   length: 1
  proto__: Array(0)
```

Nós queremos as coordenadas do nariz, então dentro de results -> dentro de index 0 ->

dentro de pose -> há nose (nariz). Escrevemos: results [0]. pose. nose

Em nose, queremos a coordenada x:



```
▼ [{...}] 🚺
 - Ø:
     ▶ keypoints: (17) [{...}, {...}, {...}, {...}, {...}, {...},
     ▶ leftAnkle: {x: 216.4462605415032, y: 301.3378774910...
     ▶ leftEar: {x: 188.69164561667637, y: 161.36775602374.
     ▶ leftElbow: {x: 268.9376267773366, y: 303.8666393324.
     ▶ leftEye: {x: 162.2530000251636, y: 154.137602465891.
     ▶ leftHip: {x: 213.40800711983127, y: 297.16762311277...
     ▶ leftKnee: {x: 209.40526075530468, y: 297.9268546690...
     ▶ leftShoulder: {x: 223.48277129625018, y: 258.861313.
     ▶ leftWrist: {x: 263.52472737518667, y: 303.070325461.
      nose: {x: 145.55414210983187 y: 178.28064153068942.
     rightAnkle: {x: 103.52699115262394, y: 299.35830692.
     ▶ rightEar: {x: 114.41234552372268, y: 175.0747362772.
     ▶ rightElbow: {x: 49.633986489814625, y: 308.25721171.
     ▶ rightEye: {x: 130.25980309436196, y: 159.9434983660.
     ▶ rightHip: {x: 102.40974590095162, y: 301.0057866224...
     ▶ rightKnee: {x: 100.08567090620073, y: 299.290570259...
     ▶ rightShoulder: {x: 83.36848592200474, y: 252.199065.
     ▶ rightWrist: {x: 42.54400660420021, y: 304.860674428.
       score: 0.30651417103431683
     ▶ __proto__: Object
   ▶ skeleton: []
   ▶ __proto__: Object
   length: 1
  __proto__: Array(0)
```

 Então o código vai ser: dentro de results -> dentro de index 0 -> dentro de pose -> dentro de nose -> coordenada x. Código:

results[0].pose.nose.x



3. Agora nós buscamos o código para obter a coordenada x de nose, então vamos atualizar a variável noseX com este código, na função **gotPoses()**.

```
function gotPoses(results)
{
   if(results.length > 0)
   {
      console.log(results);
      noseX = results[0].pose.nose.x;
}
```

P: Você pode dizer qual será o código para buscar a coordenada y do nariz?

```
results[0].pose.nose.y;
```

4. Agora nós buscamos o código para obter a coordenada y do nariz, então vamos atualizar a variável noseY com este código, na função gotPoses().

```
function gotPoses(results)
{
  if(results.length > 0)
  {
    console.log(results);
    noseX = results[0].pose.nose.x;
    noseY = results[0].pose.nose.y;
}
```



5. Agora vamos imprimir essas duas variáveis na tela do console:

```
function gotPoses(results)
{
   if(results.length > 0)
   {
      console.log(results);
      noseX = results[0].pose.nose.x;
      noseY = results[0].pose.nose.y;
      console.log("noseX = " + noseX +" noseY = " + noseY);
}
```

#### Saída na tela do console:

```
PoseNet Is Initialized!
                                                                                 <u>main.js:19</u>
► Array(2)
                                                                                 main.js:27
noseX = 300.4271958538896 noseY = 336.37838254430375
                                                                                 main.js:30
leftWristX = 402.7145948094002 rightWristX = 143.57750067236827 difference =
                                                                                main.js:36
▶ Array(2)
                                                                                 main.js:27
noseX = 299.35612525159155 noseY = 335.6596466654923
                                                                                 main.js:30
leftWristX = 373.4303239725719 rightWristX = 159.81577330397818 difference =
                                                                                main.js:36
213
                                                                                 main.js:27
noseX = 299.42421429803267 noseY = 333.9511128661693
                                                                                 main.js:30
leftWristX = 440.9891518933034 rightWristX = 150.4067573398642 difference =
                                                                                 main.js:36
                                                                                 main.js:27
▶ Array(2)
noseX = 300.05816934633907 noseY = 334.89289945735806
                                                                                 main.js:30
leftWristX = 445.36730455957195 rightWristX = 163.8111929679707 difference =
                                                                                 main.js:36
281
                                                                                 main.js:27
▶ Array(2)
noseX = 299.7428029601337 noseY = 333.90950650034824
                                                                                 <u>main.js:30</u>
leftWristX = 421.945762587802 rightWristX = 159.92287678560552 difference =
                                                                                 main.js:36
262
```

A saída será contínua, pois estamos trabalhando em tempo real.

Agora nós buscamos as coordenadas x e y do nariz, então vamos desenhar um quadrado usando essas coordenadas.



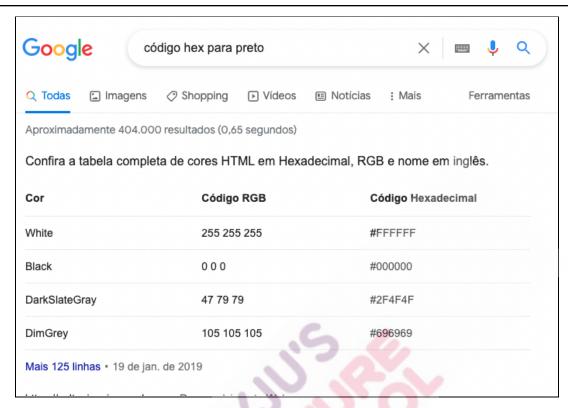
Antes de escrevermos o código para desenhar um quadrado, definiremos a cor do quadrado e a cor da borda, se não as definirmos, por padrão, o p5.js as definirá como branco e preto.

- P: Em qual função vamos escrever um código para definir a cor do quadrado e a cor da borda? Em qual função vamos escrever um código para desenhar o quadrado?
  R: Na função draw().
  - 6. fill() é uma função p5.js usada para definir a cor. Usaremos esta função para definir a cor do quadrado. Na função fill() você pode passar código RGB, código HEX, código RGBA. Aqui usamos um código HEX.

```
function draw() {
background('#969A97');
fill('#F90093');
}
```

- #F90093 : este é o código HEX da cor rosa-escuro. Não é obrigatório usar esta cor, você pode usar a cor de sua escolha.
- Para obter o código HEX de qualquer cor, basta pesquisar no google: "código HEX para nome da cor", por exemplo:





#### Agora defina a cor da borda para o quadrado.

7. **stroke()** é uma função p5.js usada para definir a cor da borda. Usaremos esta função para definir a cor da borda do quadrado. Dentro da função **stroke()** você pode passar código RGB, código HEX, código RGBA. Aqui usamos um código HEX.

```
function draw() {
background('#969A97');
  fill('#F90093');
  stroke('#F90093');
}
```

• #F90093 : este é o código HEX da cor rosa-escuro. Não é obrigatório usar esta cor, você pode usar a cor de sua escolha.

#### Agora vamos escrever o código para desenhar um quadrado na tela.

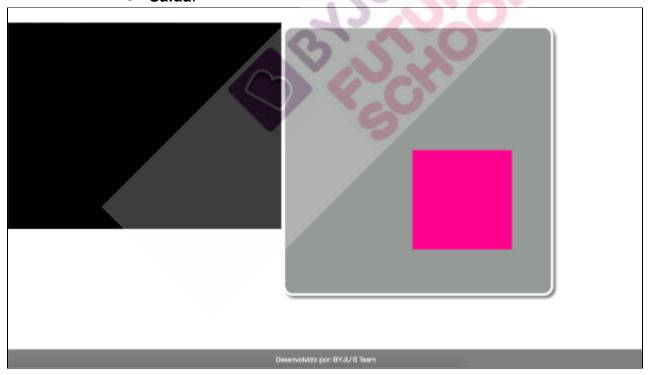
square() é uma função p5.js usada para desenhar um quadrado.
 Sintaxe de square(): square(coordenada x, coordenada y, lado);



- Então vamos colocar as coordenadas x e y do nariz, no lugar da coordenada x, coordenada y de square(), o que resultará no movimento do quadrado conforme o momento do nariz.
- E por enquanto vamos colocar um valor padrão de 100 em lado.

```
function draw() {
background('#969A97');
fill('#F90093');
stroke('#F90093');
square(noseX, noseY, 100);
}
```

- Na função square() temos 100 como o lado do quadrado, ou seja, a largura e a altura serão 100px.
- Mais tarde no código, atualizaremos o lado com um valor dinâmico (sendo a diferença entre leftWrist e rightWrist).
- Saída:



Observação para a professora: quando o aluno terminar até aqui, peça que faça um teste, e ver se está tudo funcionando bem, o quadrado está se movendo conforme o momento do nariz. O movimento do quadrado pode não ser uma imagem espelhada, porque não estamos usando



canvas para visualização da webcam.

Nossa segunda tarefa será buscar as coordenadas x do pulso esquerdo e direito e usar essas coordenadas para aumentar e diminuir o tamanho do quadrado.

A lógica por trás de aumentar e diminuir o tamanho do quadrado é:

- Buscaremos as coordenadas x do pulso esquerdo e direito e encontraremos a diferença entre eles.
- Encontrando as médias da diferença: faremos uma subtração entre as coordenadas x do pulso esquerdo e direito e usaremos esse valor de diferença para atualizar o tamanho do quadrado.

OBSERVAÇÃO: A imagem abaixo está presente na Atividade da Professora 2, para representação visual.



coordenada x do pulso direito = 50

diferença = coordenada x do pulso esquerdo - coordenada x do pulso esquerdo

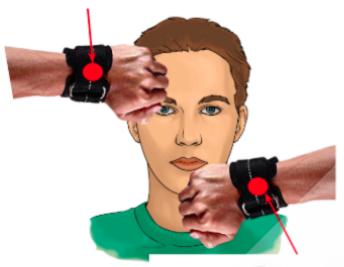
diferença = 150 - 50

diferença = 100

então o lado do quadrado será 100, por exemplo: 100 de largura e de altura



### coordenada x do pulso esquerdo = 250



coordenada x do pulso direito = 10

diferença = coordenada x do pulso esquerdo - coordenada x do pulso

esquerdo

diferença = 250 - 10

diferença = 240

então o lado do quadrado será 240, por exemplo: 240 de largura e de altura

- 1. Primeiro vamos definir 1 variável para manter a diferença entre as coordenadas x do pulso esquerdo e direito, e inicializar com um valor de 0.
- 2. Em seguida, defina 2 variáveis para manter as coordenadas x do pulso esquerdo e direito e inicialize com um valor de 0.
- 3. Defina essas variáveis no início do arquivo main.js.

```
noseX=0;
noseY=0;
difference = 0;
rightWristX = 0;
leftWristX = 0;
```



- 4. Agora busque a coordenada x do pulso da mão esquerda.
  - Execute a Atividade da Professora 1 e abra a tela do console.
  - Saída na tela do console:

PoseNet Is Initialized	main.js:17
▶ [{}]	main.js:24

Precisamos ler este objeto e buscar a coordenada x do pulso esquerdo, enquanto lemos o objeto também escreveremos o código:

5. Primeiro clique na seta para expandir:



Queremos ler os objetos de **resultados**, então primeiro vamos escrever



6. Em seguida, clique na seta ao lado de para expandir:

```
main.js:24
▶0: {pose: {...}, skeleton: Array(0)}
 length: 1
 __proto__: Array(0)
```

Clicamos no índice 0 que está no objeto de resultados para que o código seja:

```
results[0]
```

7. Em seguida, clique na seta ao lado de [ para expandir:

```
main.js:24
▼ [ {...} ] 🚺
   pose: {score: 0.25857010390866303, keypoints: Array(1...
    ▶ skeleton: []
   ▶ __proto__: Object
   length: 1
 ▶ __proto__: Array(0)
```

© 2022 - BYJU'S FutureSchool.

Nota: este documento é o copyright original da BYJU'S FutureSchool.

Não compartilhe, baixe ou copie este arquivo sem permissão.



• Então, no índice 0, clicamos no objeto de pose, então o código será:

## results[0].pose

Essa quantidade de código permanecerá a mesma sempre que quisermos acessar qualquer parte do corpo.

- Então no objeto pose há duas partes importantes <u>keypoints</u>: e 17 partes do corpo com coordenadas x e y.
- o **keypoints:** tem a mesma coisa, as 17 partes do corpo com coordenadas x e y.
- Portanto, n\u00e3o precisamos entrar em <u>keypoints</u>, porque encontraremos todas as coordenadas x e y de 17 partes do corpo fora:

```
▼ [{...}] []
 - 0:
     ▶ keypoints: (17) [{...}, {...}, {...}, {...},
     ▶ leftAnkle: {x: 216.4462605415032, y: 301.3378774910.
     ▶ leftEar: {x: 188.69164561667637, y: 161.36775602374.
     ▶ leftElbow: {x: 268.9376267773366, y: 303.8666393324...
     ▶ leftEye: {x: 162.2530000251636, y: 154.137602465891.
     ▶ leftHip: {x: 213.40800711983127, y: 297.16762311277...
     ▶ leftKnee: {x: 209.40526075530468, y: 297.9268546690...
     ▶ leftShoulder: {x: 223.48277129625018, y: 258.861313...
     ▶ leftWrist: {x: 263.52472737518667, y: 303.070325461
     ▶ nose: {x: 145.55414210983187, y: 178.28064153068942.
     ▶ rightAnkle: {x: 103.52699115262394, y: 299.35830692.
     ▶ rightEar: {x: 114.41234552372268, y: 175.0747362772...
     ▶ rightElbow: {x: 49.633986489814625, y: 308.25721171.
     ▶ rightEye: {x: 130.25980309436196, y: 159.9434983660.
     ▶ rightHip: {x: 102.40974590095162, y: 301.0057866224...
     ▶ rightKnee: {x: 100.08567090620073, y: 299.290570259...
     ▶ rightShoulder: {x: 83.36848592200474, y: 252.199065...
     ▶ rightWrist: {x: 42.54400660420021, y: 304.860674428.
       score: 0.30651417103431683
     ▶ __proto__: Object
   ▶ skeleton: []
   ▶ __proto__: Object
   length: 1
   __proto__: Array(0)
```

8. Então nós queremos as coordenadas do nariz, então dentro de results -> dentro de index 0 -> dentro de pose -> há leftWrist. Então vamos escrever: results [0] . pose . leftWrist



9. Em leftWrist, queremos a coordenada x:

```
▼ [{...}] []
 - 0:
     ▶ keypoints: (17) [{...}, {...}, {...}, {...}, {...}, {...}, {...},
     ▶ leftAnkle: {x: 216.4462605415032, y: 301.3378774910.
     ▶ leftEar: {x: 188.69164561667637, y: 161.36775602374...
     ▶ leftElbow: {x: 268.9376267773366, y: 303.8666393324.
     ▶ leftEye: {x: 162.2530000251636, y: 154.137602465891.
     ▶ leftHip: {x: 213.40800711983127, y: 297.16762311277...
     ▶ leftKnee: {x: 209.40526075530468, y: 297.9268546690.
     ▶ leftShoulder: {x: 223.48277129625018, y: 258.861313...
     ▶ leftWrist: {x: 263.52472737518667 y: 303.070325461...
     ▶ nose: {x: 145.55414210983187, y: 178.28064153068942...
     ▶ rightAnkle: {x: 103.52699115262394, y: 299.35830692.
     ▶ rightEar: {x: 114.41234552372268, y: 175.0747362772...
     ▶ rightElbow: {x: 49.633986489814625, y: 308.25721171.
     ▶ rightEye: {x: 130.25980309436196, y: 159.9434983660.
     ▶ rightHip: {x: 102.40974590095162, y: 301.0057866224...
     ▶ rightKnee: {x: 100.08567090620073, y: 299.290570259.
     ▶ rightShoulder: {x: 83.36848592200474, y: 252.199065...
     ▶ rightWrist: {x: 42.54400660420021, y: 304.860674428.
       score: 0.30651417103431683
     ▶ __proto__: Object
   ▶ skeleton: []
    __proto__: Object
   length: 1
   __proto__: Array(0)
```

10. Então o código vai ser: dentro de results -> dentro de index 0 -> dentro de pose -> dentro de results [0].pose.leftWrist.x;

lestWrist -> coordenada x. Código:

· · ·

11. Agora nós buscamos o código para obter a coordenada x de leftWrist, então vamos atualizar a variável noseX com este código, na função **gotPoses()**.



```
function gotPoses(results)
{
   if(results.length > 0)
        console.log(results);
        noseX = results[0].pose.nose.x;
        noseY = results[0].pose.nose.y;
        console.log("noseX = " + noseX +" noseY = " + noseY);
        leftWristX = results[0].pose.leftWrist.x;
}
```

P: Você pode dizer qual será o código para buscar a coordenada x de rightWrist?

```
results[0].pose.rightWrist.x;
```

12. Agora nós buscamos o código para obter a coordenada x de rightWrist, então vamos atualizar a variável rightWristX com este código, na função gotPoses().



13. Agora vamos fazer uma subtração entre as coordenadas x do pulso esquerdo e direito, e atualizar a variável difference (diferença) com esse valor.

```
function gotPoses(results)
{
   if(results.length > 0)
{
     console.log(results);
     noseX = results[0].pose.nose.x;
     noseY = results[0].pose.nose.y;
     console.log("noseX = " + noseX +" noseY = " + noseY);

   leftWristX = results[0].pose.leftWrist.x;
   rightWristX = results[0].pose.rightWrist.x;

   difference = leftWristX - rightWrist;
}
```



## OBSERVAÇÃO: A imagem abaixo está presente na Atividade da Professora 2, para representação visual.

• Mas, como você pode ver, as coordenadas x de leftWrist e rightWrist têm muitos decimais:

```
rightWrist: {x: 263.52472737518667,
rightWrist: {x: 42.54400660420021,
```

- Então, quando fazemos uma subtração entre as coordenadas x de leftWrist e rightWrist, obteremos um resultado que terá muitos decimais. (variável difference contém um resultado da subtração entre as coordenadas x de leftWrist e rightWrist)
- Não queremos muitos decimais nos resultados, então vamos remover todos os decimais desta variável **difference** usando a função floor().
- A função floor() é uma função p5.js usada para remover todos os decimais e reduzir o valor. Por exemplo:
  - Se o valor for 45,67937: então o resultado será 45
  - Se o valor for 45,43553: então o resultado será 45
  - Se o valor for 45,10279: então o resultado será 45
- Como conclusão, as funções floor() removem todos os decimais.
- Sintaxe de floor() floor(variable\_name)
- 14. Então vamos adicionar a função floor() ao código:



15. Vamos imprimir essas três variáveis no console (leftWristX, rightWristX e difference).

```
function gotPoses(results)
{
   if(results.length > 0)
   {
      console.log(results);
      noseX = results[0].pose.nose.x;
      noseY = results[0].pose.nose.y;
      console.log("noseX = " + noseX +" noseY = " + noseY);

   leftWristX = results[0].pose.leftWrist.x;
   rightWristX = results[0].pose.rightWrist.x;
   difference = floor(leftWristX - rightWristX);

console.log("leftWristX = " + leftWristX + " rightWristX = " + rightWristX + " difference = " + difference);
}
```

Saída na tela do console:

```
PoseNet Is Initialized!
                                                                                main.js:19
                                                                               main.js:27
▶ Array(2)
noseX = 300.4271958538896 noseY = 336.37838254430375
                                                                                main.js:30
leftWristX = 402.7145948094002 rightWristX = 143.57750067236827 difference =
                                                                               main.js:36
259
                                                                                main.js:27
▶ Array(2)
noseX = 299.35612525159155 noseY = 335.6596466654923
                                                                                main.js:30
leftWristX = 373.4303239725719 rightWristX = 159.81577330397818 difference =
                                                                               main.js:36
213
▶ Array(2)
                                                                                main.js:27
noseX = 299.42421429803267 noseY = 333.9511128661693
                                                                                main.js:30
leftWristX = 440.9891518933034 rightWristX = 150.4067573398642 difference =
                                                                               main.js:36
                                                                                main.js:27
▶ Array(2)
noseX = 300.05816934633907 noseY = 334.89289945735806
                                                                                main.js:30
leftWristX = 445.36730455957195 rightWristX = 163.8111929679707 difference =
                                                                               main.js:36
281
                                                                                main.js:27
▶ Array(2)
noseX = 299.7428029601337 noseY = 333.90950650034824
                                                                                main.js:30
leftWristX = 421.945762587802 rightWristX = 159.92287678560552 difference =
                                                                               main.js:36
262
```

A saída será contínua, pois estamos trabalhando em tempo real.



16. Agora temos a diferença entre leftWrist e rightWrist, então usaremos essa diferença para atualizar os lados (largura e altura) do quadrado. Então atualize o **lado**, na função **square()**.

```
function draw() {
background('#969A97');
fill('#F90093');
stroke('#F90093');
square(noseX, noseY, difference);
}
```

Lembre-se que na aula 123 no código HTML, nós definimos uma tag span vazia com id "square\_side"

<span id="square\_side"></span>
. Isso foi definido para manter os novos lados (largura e altura) do quadrado.

Então agora calculamos o novo lado (que é a diferença entre o pulso esquerdo e o pulso direito).

- 17. Assim, atualizaremos a tag span com o novo lado (que é a diferença entre leftWrist e rightWrist). Adicione também um texto.
- 18. E esse código de atualização da tag span com o novo lado (que é a diferença entre leftWrist e rightWrist) virá dentro da função **draw()**, porque o lado do quadrado está mudando continuamente, então precisamos atualizar continuamente a tag span com o novo lado.

```
function draw() {
background('#969A97');

document.getElementById("square_side").innerHTML = "Largura e altura serão = " + difference +"px";
fill('#F90093');
stroke('#F90093');
square(noseX, noseY, difference);
}
```



#### Explicando o código acima:

- document.getElementById("square\_side").innerHTML: isso significa que estamos nos referindo a um elemento HTML que possui um id "square\_side" e o atualizando.
- Então estamos escrevendo o texto "Largura e altura serão = "

```
document.getElementById("square_side").innerHTML = "Largura e altura serão = "
```

• Então temos que concatenar a variável difference com a string para usarmos o sinal "+".

```
document.getElementById("square_side").innerHTML = "Largura e altura serão = " +
```

• Depois colocamos a variável difference.

```
document.getElementById("square_side").innerHTML = "Largura e altura serão = " + difference
```

• Então temos que concatenar o "px" com a variável difference para usarmos o sinal "+".

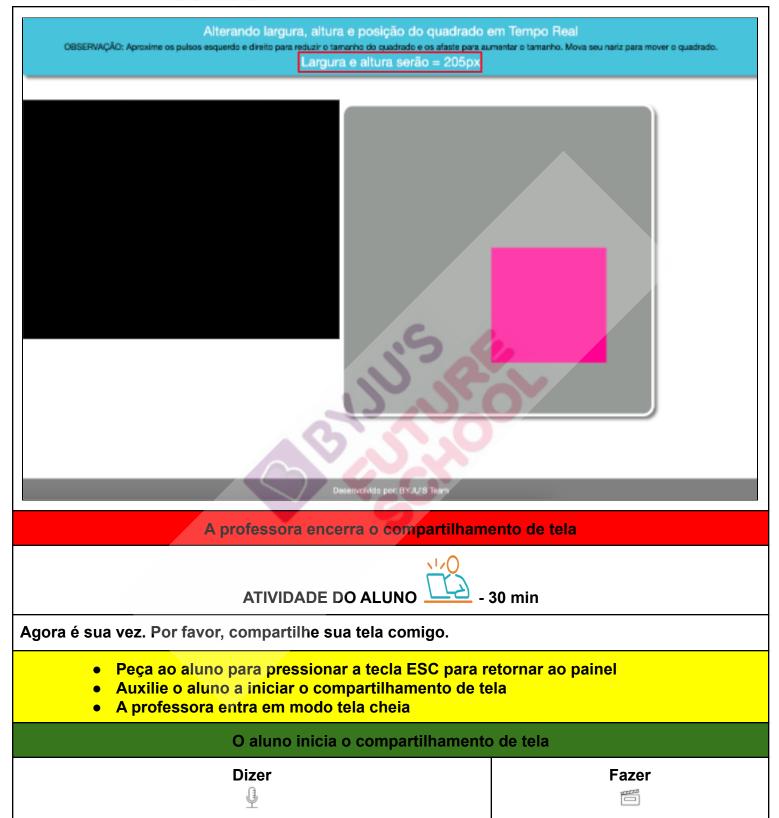
```
document.getElementById("square_side").innerHTML = "Largura e altura serão = " + difference +
```

Como o lado (largura e altura) do quadrado está em pixels, então vamos escrever "px".

```
document.getElementById("square_side").innerHTML = "Largura e altura serão = " + difference +"px";
```

Resultado:







\*Observação: Por favor, siga os seguintes passos.

Atividade do Aluno 1 - DIAGRAMA DO CÓDIGO

Atividade do Aluno 2 - HABILITAR WEBGL

Referência para Atividade do Aluno 1 - REFERÊNCIA DA FUNÇÃO FILL() DE p5.js

Referência para Atividade do Aluno 2 - REFERÊNCIA DA FUNÇÃO SQUARE() DE p5.js

O aluno deve continuar a programação JS no arquivo main.js.

Se o tempo permitir, incentive o aluno a projetar livremente.

OBSERVAÇÃO PARA PROFESSORA -

Peça ao aluno para hospedar o site. As etapas são mencionadas na Referência para Atividade do Aluno 3.

Ele deve copiar o link do GitHub e colar no campo Enviar link no painel do aluno e clicar no botão Enviar.

Teste o app usando este link em seu telefone.

Continue a programação JS no arquivo main.js.

Agora vamos fazer upload de todos os arquivos para hospedar seu site no GitHub.

As etapas para hospedar o site são mencionadas na Referência para Atividade do Aluno 3.

Copie o link do GitHub e cole no campo de Enviar Link no painel do aluno, e clique no botão Enviar.

Teste o app usando este link em seu telefone.

A professora auxilia o aluno a encerrar o compartilhamento de tela

**SESSÃO DE ENCERRAMENTO - 5 min** 

A professora inicia a apresentação de slides

do slide 16 ao 18.

Consulte as notas da oradora e siga as instruções em cada slide.



Você receberá tiradas de chapéu pelo excelente trabalho!

Aplauda o aluno e dê tiradas de chapéu.

(Dê, no mínimo, 2 tiradas de chapéu

Pressione o ícone de Tirada de Chapéu para *Resolvendo Atividades Criativamente*.



Pressione o ícone de Tirada de Chapéu para *Ótima Pergunta*.



Pressione o ícone de Tirada de Chapéu para "*Você se Concentrou*".



#### **DICAS E SUGESTÕES DO PROJETO - 5 min**

A professora inicia a apresentação de slides



dos slides 20 e 21.

Consulte as notas da oradora e siga as instruções em cada slide.

Dizer



**Fazer** 



#### **MANIPULADOR DE FONTE - 3**

#### Objetivo do Projeto:

Na aula de hoje, completamos o código para inicializar a posenet e executar o modelo.

Neste projeto você tem que escrever código para acessar a webcam, criar uma tela, inicializar e executar o modelo posenet.

Observação: Você pode atribuir o projeto ao aluno na própria aula, clicando no botão Atribuir Projeto que está disponível na guia Projetos.

Abra o link da **Solução do Projeto** e demonstre o projeto ao aluno.

História:

© 2022 - BYJU'S FutureSchool.

Nota: este documento é o copyright original da BYJU'S FutureSchool. Não compartilhe, baixe ou copie este arquivo sem permissão.



Na escola Santa Fada, os alunos não gostavam de ler o quadro de avisos e geralmente perdiam as mensagens importantes que os professores/diretores tinham que transmitir. Então o diretor decidiu fazer um divertido quadro de avisos para os alunos onde um aluno pode aumentar ou diminuir o tamanho da fonte. Se eles não puderem ver o aviso escrito em letras pequenas, eles podem usar os pulsos para aumentar a fonte e ler o que está escrito.

O diretor gostou do seu projeto anterior, que você fez a programação para acessar a webcam e o modelo posenet, e agora ele está animado para o projeto completo.

Eu sei que você vai se sair muito bem.

Até breve!

#### A Professora Para ø Compartilhamento de Tela

#### ATIVIDADE ADICIONAL

A professora inicia a apresentação de slides



do slide 22 ao 26.

#### ATIVIDADE ADICIONAL DO ALUNO



#### O Aluno Inicia o Compartilhamento de Tela

Dizer

**Fazer** 



Para a solução de todas as Atividades Adicionais, abra a Atividade da Professora - 4 e navegue até o número de aula C125.

Atividade Adicional 1 -

Execute a Atividade do Aluno-3 do painel

AS TAREFAS e DICAS são mencionadas no próprio website.

Atividade Adicional 2 -

Execute a Atividade do Aluno-4 do painel

AS TAREFAS e DICAS são mencionadas no próprio website.

© 2022 - BYJU'S FutureSchool.

Nota: este documento é o copyright original da BYJU'S FutureSchool. Não compartilhe, baixe ou copie este arquivo sem permissão.



Atividade Adicional 3 -

Execute a Atividade do Aluno-5 do painel

AS TAREFAS e DICAS são mencionadas no próprio website.

Atividade Adicional 4 -

Execute a Atividade do Aluno-6 do painel

AS TAREFAS e DICAS são mencionadas no próprio website.

Atividade Adicional 5 -

Execute a Atividade do Aluno-7 do painel

AS TAREFAS e DICAS são mencionadas no próprio website.

#### A Professora Orienta o Aluno a Parar o Compartilhamento de Tela

A professora termina a apresentação de slides

no slide 27.

Consulte as notas da oradora e siga as instruções em cada slide.

### A professora clica em

× Terminar Aula

Atividade	Nome da Atividade	Links
Atividade da Professora 1	LINK DO SITE	https://c-amilarorigues.github.io/ADV-V2-Desenho-Tempo-Real/
Atividade da Professora 2	DIAGRAMA DO CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/ ee1ecfc0-02c2-4ae4-bb59-ca1067490d84.pdf
Atividade da Professora 3	CÓDIGO FONTE	https://drive.google.com/drive/folders/192lv-Rk MO2dpkm2TjNg8SdzFecQn93Jj?usp=sharing
Atividade da Professora 4	SOLUÇÃO DAS ATIVIDADES ADICIONAIS	https://docs.google.com/spreadsheets/d/e/2PA CX-1vSG0V0IQyGx7Zo7VJgaWAHYDpalldCZ -RepABc_slelct65DaM-R9YF0vCbS5JFdsF3M EFQ-WLslsRc/pubhtml
Atividade da Professora 5	HABILITAR <b>WEBGL</b>	https://s3-whjr-curriculum-uploads.whjr.online/ddf97684-bcce-4c74-9384-46bc4741f2ab.pdf
Atividade do Aluno 1	DIAGRAMA DO CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/



		ee1ecfc0-02c2-4ae4-bb59-ca1067490d84.pdf
Atividade do Aluno 2	HABILITAR <b>WEBGL</b>	https://s3-whjr-curriculum-uploads.whjr.online/ddf97684-bcce-4c74-9384-46bc4741f2ab.pdf
Referência para Atividade do Aluno 1	REFERÊNCIA DA FUNÇÃO FILL() DE p5.js	https://p5js.org/reference/#/p5/fill
Referência para Atividade do Aluno 2	REFERÊNCIA DA FUNÇÃO SQUARE() DE p5.js	https://p5js.org/reference/#/p5/square
Referência para Atividade do Aluno 3	REFERÊNCIA PARA HOSPEDAR UM SITE NO GITHUB	https://s3-whjr-curriculum-uploads.whjr.online/d4537bb0-6e4f-4843-9069-3d1fa70449ae.pdf
Solução do Projeto	MANIPULADOR DE FONTE - 3	https://s3-whjr-curriculum-uploads.whjr.online/ 00c1645e-bddb-48c9-a8c7-6a1651791f4b.pdf
Link de Recurso Visual Referência da Professora	Recurso Visual COM NOTAS	https://s3-whjr-curriculum-uploads.whjr.online/ bca64f4d-089c-4ae8-b23b-532aabb88f77.html
Link de Recurso Visual Referência da Professora	Recurso Visual SEM NOTAS	https://s3-whjr-curriculum-uploads.whjr.online/f 37ad92d-0b0f-4dce-808c-c75db32de200.html
Link Questionário de Aula Referência da Professora	Questionário de Aula	https://s3-whjr-curriculum-uploads.whjr.online/ b8027404-ccc8-4f7c-a26d-a3e1fddf6b3f.pdf