

Tópico	JOGO COM IA		
Descrição da Aula	A criança revisa e reaplica todos os conceitos de <i>machine learning</i> (aprendizado de máquina) para aprimorar um jogo popular, fazendo com que funcione com redes neurais. Assim, aprenderá a aplicar IA no mundo real.		
Aula	ADV-C138		
Duração da Aula	55 min		
Objetivo	<ul style="list-style-type: none"> ● Iniciar o design UI do jogo Mario 		
Recursos Necessários	<ul style="list-style-type: none"> ● Recursos da Professora <ul style="list-style-type: none"> ○ Use suas credenciais de login do Gmail; ○ Fone de ouvido com microfone; ○ Bloco de notas e caneta; ○ Webcam. ● Recursos do Aluno <ul style="list-style-type: none"> ○ Use suas credenciais de login do Gmail; ○ Fone de ouvido com microfone (opcional); ○ Bloco de notas e caneta; ○ Webcam. 		
Estrutura da Aula	Introdução Atividade Dirigida pela Professora Atividade Dirigida pelo Aluno Encerramento		5 min 15 min 35 min 5 min
Passos da Aula	Dizer	Fazer	
Passo 1: Introdução (5 min)	Nas últimas aulas, trabalhamos com a detecção de objetos e criamos alguns apps com ela. <ul style="list-style-type: none"> ● Desenvolvemos um app web que realiza a detecção de objetos em uma imagem 	Peça para o aluno ativar o modo Tela cheia .	

- estática de um cachorro e um gato;
- Desenvolvemos um App web que realiza a detecção de objetos a partir da visualização ao vivo da webcam;
 - Desenvolvemos um App web que realiza a detecção de objetos em um vídeo.

P: Você se lembra de qual modelo utilizamos para a detecção de objetos?

R: Utilizamos o modelo cococssd.

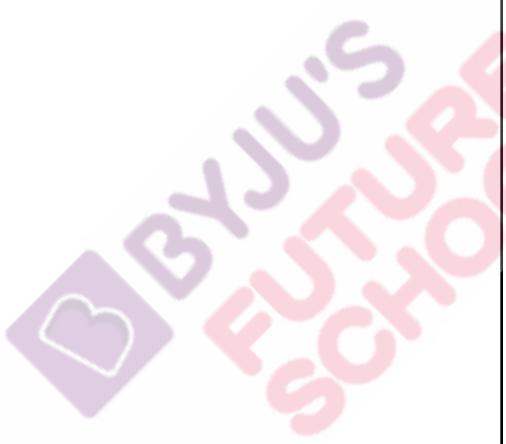
P: O que o modelo cococssd nos retorna quando lhe fornecemos uma imagem/vídeo?

R: Ele nos retorna:

- O nome dos objetos detectados e sua *confidence* (precisão);
- As coordenadas x e y dos objetos detectados;
- A *width* (largura) e *height* (altura) dos objetos detectados.

P: Em que lugar podemos utilizar o nome, a *confidence*, as coordenadas x e y, a *width* e a *height* que recebemos do modelo?

R: Utilizamos esses valores para desenharmos os retângulos ao redor dos objetos detectados e

	<p>posicionarmos as <i>labels</i> (etiquetas) e <i>confidence</i> perto dos objetos.</p> <p>Agora, seguiremos em direção à última fase do módulo IA, no qual desenvolveremos uma IA de Jogo.</p>	
A professora inicia o compartilhamento de tela		
Passo 2: Atividade Dirigida pela Professora (15 min)		<p>Se você fizer o download do código completo da Atividade da Professora 4 e quiser testar em seu sistema,</p> <p>CERTIFIQUE-SE DE CLICAR NO BOTÃO GO LIVE DO VISUAL STUDIO. COM ISSO OS ARQUIVOS SERÃO EXECUTADOS NO LIVE SERVER DO VISUAL STUDIO</p>  <p>Utilizaremos arquivos de imagem e áudio, no entanto, a biblioteca p5.js não nos permite executar arquivos de imagem e áudio do sistema local, portanto, precisaremos executá-los em um SERVER.</p> <p>Por favor, siga o fluxo desta aula</p> <p>Demonstração do jogo Mario Atividade da Professora 1</p> <p>Demonstração do jogo IA Mario Atividade da Professora 2</p> <p>Explique os códigos HTML e CSS.</p> <p>Deixe o aluno programar em HTML e CSS.</p>

		Continue com as Atividades Adicionais, caso haja tempo.
--	--	---

Primeiro, executarei um jogo básico no modelo Mario:

OBSERVAÇÃO: Execute a Atividade da Professora 1.



Pressione qualquer tecla para iniciar o jogo. Tente jogar por 1 minuto, com as teclas direcionais, para mostrar o jogo ao aluno.

Agora, demonstrarei uma este jogo com uma IA.

OBSERVAÇÃO: Execute a Atividade da Professora 2

Mostre a UI -

1. Título e Imagem:

IA MARIO

Instruções



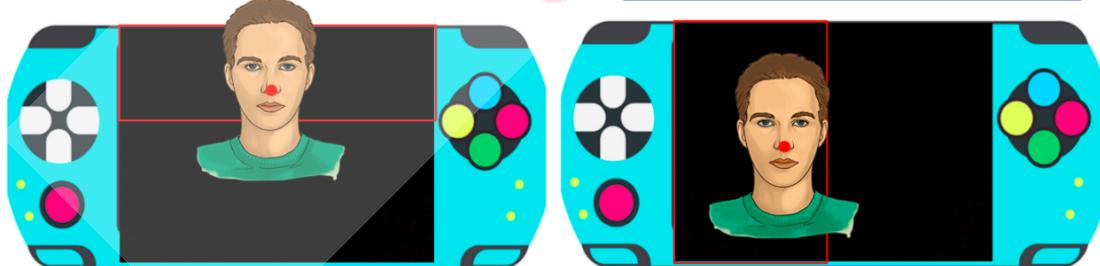
Instruções

2. Pressione o botão de instruções para mostrar como jogar

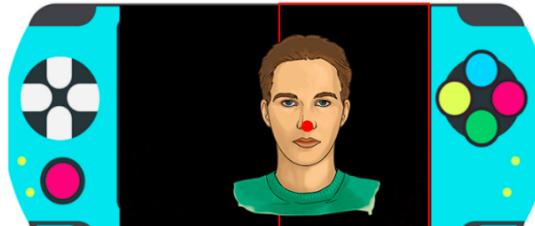
Instruções

Coloque seu nariz na parte superior do vídeo para pular

Coloque seu nariz no lado esquerdo do vídeo para mover para a direita, pois a visualização da webcam não é uma imagem espelhada



Coloque seu nariz no lado direito do vídeo para mover para a esquerda, pois a visualização da webcam não é uma imagem espelhada



Após mostrar as instruções, feche o pop-up.

3. Botão “Jogar”, canvas do jogo e console para exibir a visualização ao vivo da webcam.



4. Créditos.

Código fonte: [Linuk](#) || Créditos: Mario

- Utilizaremos o código criado por Linuk. Por isso, criamos a seção de créditos para mencionar seu nome Código fonte: [Linuk](#) ||
- Visto que replicaremos o jogo “Mario”, precisaremos mencioná-lo nos créditos || Créditos: Mario
- Ao longo da aula, explicarei quem é Linuk, e por que utilizaremos seu código.



Após mostrar a UI, pressione o botão “Jogar”

Talvez o jogo demore a iniciar. Portanto, espere.



Após iniciado, o status do jogo será atualizado para:

E o jogo iniciará.

Leia as instruções, mencionadas no website, sobre como jogar.

Jogue e mostre ao aluno de que modo o Mario se move da esquerda para a direita e pula de acordo com a movimentação do seu nariz.

Você acabou de ver o jogo Mario com uma IA. Empolgado para desenvolver esse jogo?

P: Após ver a IA do jogo, você pode adicionar qual modelo será utilizado?

R: O modelo PoseNet.

P: Como podemos dizer que o modelo Pose Net é utilizado neste jogo com IA?

R: Visto que a movimentação do Mario ocorre de acordo com nosso nariz, podemos dizer que o modelo Pose Net é utilizado. Pois esse retorna as coordenadas do nariz e, a partir delas, move o personagem.

Há diversas plataformas de código aberto em que programadores e desenvolvedores podem publicar seus projetos, pesquisas e códigos.

No mundo da programação, nem todos os desenvolvedores escrevem seus códigos do zero. Muitos utilizam códigos de plataformas *open source* (código aberto), analisam a lógica por trás para, então, fazerem as modificações necessárias e o integrarem ao *software*. Em alguns casos, desenvolvedores utilizam API de terceiros e executam os mesmos passos da utilização de um código aberto. Você não precisa escrever um código do zero todas as vezes. Se você tiver o conhecimento e as habilidades para compreender o código de outra pessoa e realizar as alterações necessárias você poderá utilizar códigos abertos para suas criações.

No entanto, para isso, você precisa de muita experiência e habilidades, como:

1. Conseguir ler e entender a lógica do código de outra pessoa.
2. Realizar as alterações necessárias e integrar trechos de código ao *source* que você for utilizar.

Logo, nesta última fase do desenvolvimento IA, faremos isso. Não criaremos o jogo do zero, mas faremos o download do código fonte, criado por Linuk, e adicionaremos o código IA ao jogo.

O primeiro jogo Mario que mostrei, controlado pelas teclas direcionais, foi desenvolvido por Linuk. O segundo jogo que mostrei será o resultado de nossas modificações no código de Linuk, ao integrarmos a IA.

Na próxima aula, compartilharei o link do GitHub de Linuk, para que você possa ver a página/perfil dele. Alguns dos benefícios de utilizar o código de outra pessoa, e adicionar seu nossos trechos de código, são:

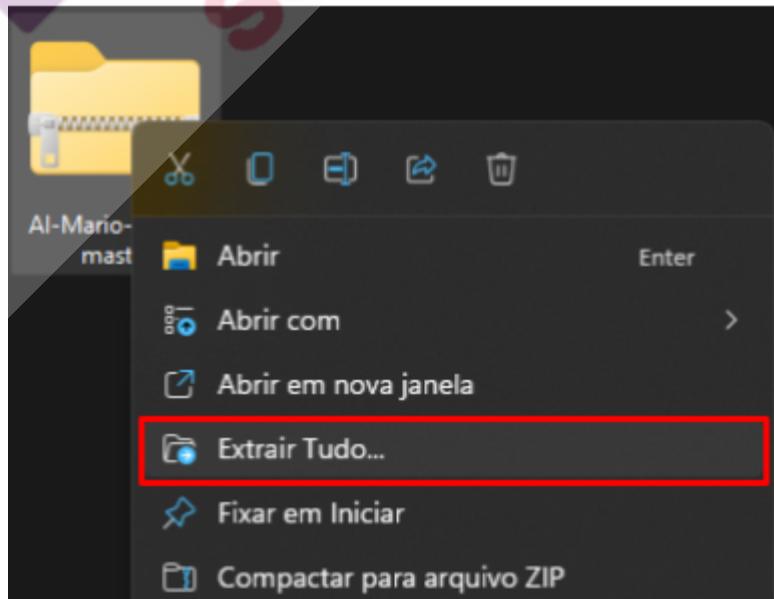
- Você terá a experiência de trabalhar com códigos grandes e complexos
- Você aprenderá a ler e compreender o código escrito por outra pessoa. Além disso, aprenderá a manipular a lógica do código.

Se você conseguir entender como utilizar o código de outra pessoa, e for capaz de realizar as alterações necessárias, então, você conseguirá utilizar códigos de qualquer plataforma *open source* e aplicar as modificações que desejar para desenvolver atividades complexas.

Aqui, finalizamos a demonstração. Portanto, começaremos a trabalhar neste código.

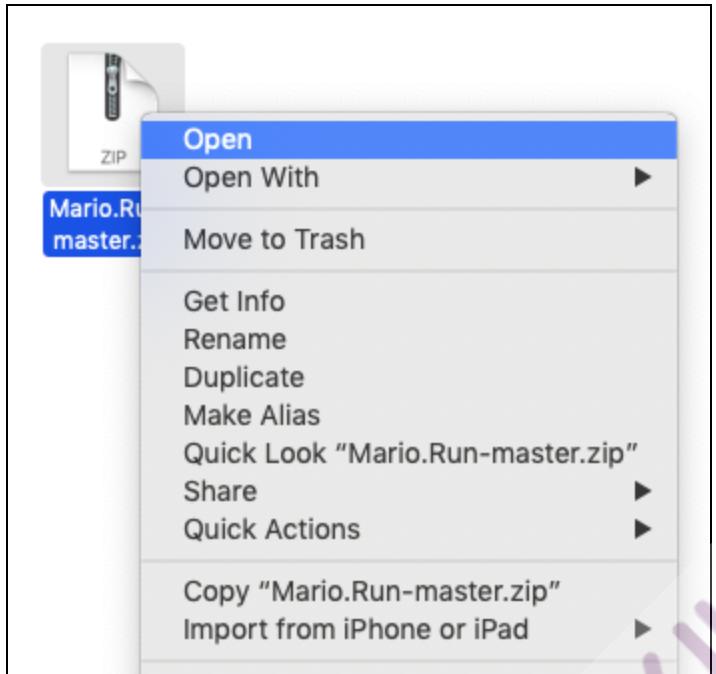
Antes de iniciar a atividade, você deverá fazer o download da pasta [Mario-Game.zip](#) da [Atividade do Aluno 1](#). A pasta [Mario-Game.zip](#) é uma pasta compactada, portanto, será necessário descompactá-la para obter seus arquivos. Após descompactar a pasta, será criada uma outra pasta com o nome [Mario-Game](#).

OBSERVAÇÃO: No Windows: Clique com o botão direito do mouse sobre o arquivo .zip e



selecione a opção “Extrair tudo” . Após a extração, uma pasta será criada, que conterá todos os arquivos e pastas.

No Mac: clique com o botão direito do mouse sobre a pasta e selecione a opção Abrir -



. Após a extração, uma pasta será criada, que

conterá todos os arquivos e pastas.

Essa pasta contém:



- **Pasta imgs:** Essa pasta possui todas as imagens necessárias para o jogo;
- **Pasta lib:** Essa pasta possui todos os arquivos necessários da biblioteca p5.js para o jogo;
- **Pasta script:** Essa pasta possui todo o código JS necessários para o jogo;
- **index.html:** Possui alguns trechos de código HTML. Você deve completar o código;
- **style.css:** É um arquivo vazio. Você deve escrever o código;
- **main.js:** Possui alguns trechos de código JS. Nas próximas aulas você começará a escrever o código neste arquivo. Este é o **primeiro** arquivo JS importante para o jogo;

- **characters_environment.js**: Possui todo o código JS do jogo. Nas próximas aulas, você começará a escrever o código neste arquivo. Este é o **segundo** arquivo JS importante para o jogo;
- **Alguns arquivos de áudio** que adicionaremos ao código durante as próximas aulas;
- **Algumas imagens** que adicionaremos ao código durante as próximas aulas.

Trabalharemos nos arquivos index.html, style.css e main.js para tornarmos esse jogo Mario em um Mario com IA.

NÃO ALTERE OS ARQUIVOS. UMA PEQUENA MODIFICAÇÃO PODE CORROMPER O CÓDIGO. PRECISAMOS TER MUITO CUIDADO AO TRABALHAR NESTE CÓDIGO, POIS ELE É MUITO EXTERNO. SE UM ERRO OCORRER, SERÁ MUITO TRABALHOSO CORRIGI-LO.

Começaremos pela UI do jogo.

O arquivo **index.html** possui o código HTML a seguir.

OBSERVAÇÃO: A imagem abaixo também está presente na [Atividade da Professora 3 para representação visual](#).

```

<!DOCTYPE html>
<html>
<head>
    <script src="https://unpkg.com/ml5@0.4.3/dist/ml5.min.js"></script> → Link ml5.js

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

    <title>p5.play.Mario</title>
    <!-- link p5.js e seus complementos como p5.dom.js ou p5.sound.js -->
    <script src="lib/p5.min.js" type="text/javascript"></script>
    <!-- link p5.play.js -->
    <script src="lib/p5.play.js" type="text/javascript"></script>
    <!-- link p5.dom.js -->
    <script src="lib/p5.dom.js" type="text/javascript"></script> → Links da biblioteca p5.js  

    necessários para o jogo

    <!-- link para arquivo javascript -->
    <script src="scripts/sprites.js" type="text/javascript"></script> → Arquivo JS importado para  

    carregar todas as imagens  

    utilizadas no jogo

    <script src="main.js" type="text/javascript"></script> → Link do nosso arquivo JS  

    (primeiro arquivo JS  

    importante para o jogo)

    <script src="characters_environment.js" type="text/javascript"></script> → Segundo arquivo JS  

    importante para o jogo

    <link rel="stylesheet" type="text/css" href="style.css"> → Link do nosso  

    arquivo de estilo

    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.0.0/addons/p5.sound.min.js"></script> → Biblioteca p5.js de som

</head>
<body>
    </body>
</html>

```

O arquivo **index.html** possui:

- Link para a biblioteca ml5.js;
- Classes Bootstrap;
- Link para a biblioteca p5.js, necessária para o jogo;
- Link para o arquivo sprites.js: Este arquivo JS é utilizado para carregar todas as imagens necessárias para o jogo;
- Link para nosso arquivo JS: Este será o primeiro arquivo JS importante para o jogo;
- Link para nosso arquivo characters_environment.js: Este será o segundo arquivo JS importante para o jogo;
- Link para nosso style sheet;
- Link para a biblioteca de som de p5.js.

Agora, começaremos com a adição dos elementos HTML no [index.html](#).

Em HTML, adicionaremos o código a seguir:

1. Primeiro, adicione a imagem de fundo da página.

P: Para adicionar a imagem de fundo, à página inteira, utilizaremos o **atributo background** dentro de qual elemento?

R: Para adicionar a imagem de fundo à página web, precisaremos adicionar um **atributo background** dentro da tag body.

```
<body background="background.jpg">
```

- Escreva um **atributo background** dentro da tag body e, dentro dele, mencione o nome da imagem **background.jpg**. Utilize o mesmo nome da imagem, pois essa está presente na pasta [Mario-Game](#) que você descompactou.

Saída:



2. Adição da tag center:

```
<body background="background.jpg">
<center>
</center>
</body>
```

- Escreveremos todo o nosso código HTML dentro dessa tag center, pois queremos que tudo esteja no centro da página.

- Então, adicionaremos uma div para conter o título e as instruções na base da página. Além disso, forneceremos algumas classes Bootstrap e nossa class nela.

```
<body background="background.jpg">
  <center>
    <div class="btn btn-primary heading">
      | </div>
    </center>
  </body>
```

- btn**: É uma class bootstrap que acionará *padding* (preenchimento) e *margin* (margem) a essa div para que tenha uma aparência melhor.
- btn-primary**: É uma class Bootstrap que adicionará a cor azul escura ao fundo e à borda da tag div.
- heading**: É a nossa class e forneceremos estilo a ela em [style.css](#).

Estilos em [style.css](#):

```
.heading
{
  width: 100%;
  text-align: left;
}
```

- Definir a *width* para 100%, para que esta heading cubra, horizontalmente, toda a tela.
- Definir *text-align* (alinhamento do texto) para *left* (à esquerda), para que o texto de nossa heading se inicie à esquerda da tela.

- Agora, adicione uma tag h3 dentro da div criada acima. Essa tag h3 conterá o texto da heading.

```
<body>
  <center>
    <div class="btn btn-primary heading">
      | <h3>IA MARIO</h3>
    </div>
  </center>
</body>
```

Estilo em `style.css`:

5. Visto que essa tag h3 está dentro da div cuja class é heading, podemos escrever:

```
.heading h3{  
    display: inline;  
}
```

- **display:inline:** Essa propriedade é utilizada quando queremos que os elementos apareçam em uma linha.
 - Você pôde ver que, no website, o título e as instruções aparecem na base da página e na mesma linha. Portanto, para alinhar o título com o botão, precisamos definir a propriedade *display* (exibição) da tag h3 para *inline* (mesma linha), para nos ajudar a manter o título e botão na mesma linha.

Saída:

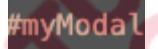
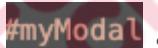


6. Agora, adicionaremos à div, criada acima, um botão para as instruções. Você pôde ver na demonstração que quando pressionamos o botão de instruções, um pop-up aparece. Você deve se lembrar que utilizamos a class Bootstrap modal, quando criamos um Blog. Logo, utilizaremos a mesma class.

```
<div class="btn btn-primary heading">
  <h3>IA MARIO</h3>
  <button class="btn btn-primary" data-toggle="modal" data-target="#myModal">Instruções</button>
</div>
```

- Adicione uma tag *button* (botão) 
- Em seguida, adicione a class Bootstrap 
- Depois, adicionaremos . **data-toggle** é utilizada para abrir o **modal/pop-up**. Com a adição dessa class, ao pressionarmos o botão, um pop-up aparecerá
- Então, adicionamos um *target* (alvo):

```
<button class="btn btn-primary" data-toggle="modal" data-target="#myModal">
```

- **data-target**: deve conter a ID do elemento HTML que deve ser aberto quando o botão for pressionado. Escreva  dentro de **data-target**. Nas próximas aulas, definiremos uma div com essa ID . Com isso, quando o botão for pressionado, a div com a ID  será aberta.
 - **data-target** funciona do mesmo modo que "target" em anchor, que já aprendemos. Ela nos leva ao elemento específico cuja id foi mencionada.
- Em seguida, forneça o texto a esse botão e feche a tag *button*:

```
<button class="btn btn-primary" data-toggle="modal" data-target="#myModal">Instruções</button>
```

Estilo em **style.css**

7. Como essa tag *button* está dentro de uma div cuja class é *heading*, podemos escrever:

```
.heading button{
  float: right;
  border: 1px solid #white;
}
```

- **float:** Essa propriedade é utilizada quando queremos que os elementos se movam para a esquerda ou direita da página.
- Queremos que esse botão “Instruções” esteja à direita da página. Para isso, utilizaremos a propriedade **float:right** nesse botão.

P: O que resulta **border:1px solid white?**

R: Resulta em uma borda branca de 1px.

Saída:



8. Agora, adicionaremos uma imagem grande após a div do título, e forneceremos nossa class.

```
<h3>IA MARIO</h3>
<button class="btn btn-primary" data-toggle="modal" data-target="#myModal">Instruções</button>
</div>


```

- Dentro do **atributo src**, mencione o nome da imagem **mario.jpg**. Escreva o mesmo nome da imagem, pois ela está presente na pasta **Mario-Game** que você descompactou.

Estilo em **style.css**.

```
.bigImage
{
    width: 100%;
}
```

Queremos que essa imagem cubra toda a largura da tela, por isso utilizamos **width:100%**.

Saída:

IA MARIO

Instruções



9. Adicione duas tags **br**.

P: Por que utilizamos **tags br?**

R: Para fornecermos quebras de linha.

```
<h3>IA MARIO</h3>
<button class="btn btn-primary" data-toggle="modal" data-target="#myModal">Instruções</button>
</div>


<br><br>
```

- Adicionamos essas duas tags **br** para que haja uma quebra de linha entre a tag **img** e a próxima tag (botão “Jogar”).

10. Adicione um botão “Jogar”. Esse botão será utilizado para iniciar o jogo. Além disso, forneceremos algumas classes Bootstrap e uma ID a essa tag **button**.

```
</div>


<br><br>
<button class="btn btn-success" onclick="startGame()" id="start">Jogar</button>
```

- **btn**: É uma class Bootstrap que adiciona preenchimento e margem a essa div, para que fique com uma aparência melhor;
- **btn-success**: É uma class Bootstrap que adiciona a cor verde clara ao fundo e às bordas da tag.
- **onclick = "startGame()"**: Chama a função startGame() quando o botão for pressionado. Essa função, será utilizada para iniciar o jogo. Definiremos ela na próxima aula.
- **id="start"**: Utilizaremos essa id para fornecer estilos em **style.css**.

Estilos em **style.css**:

```
#start
{
  font-size: 22px;
}
```

- Aumentamos o tamanho da fonte.

Saída:



11. Adicione duas tags br.

```

<br><br>
<button class="btn btn-success" onclick="startGame()" id="start">Jogar</button>
<br><br>
```

- Adicionamos essas duas tags **br** para que haja uma quebra de linha entre o botão “Jogar”

e a próxima tag h3 que será utilizada para conter o status do jogo.

12. Adicione uma tag h3 para conter o status do jogo. Além disso, forneça algumas classes Bootstrap e ID a ela.

```

<br><br>
<button class="btn btn-success" onclick="startGame()" id="start">Jogar</button>
<br><br>
<h3 id="status" class="btn btn-warning"></h3>
```

id="status": Essa id será utilizada para referir a essa tag h3 em JS enquanto atualiza o status do jogo. Além disso, com essa **id**, podemos fornecer estilos em **style.css**:

- **btn**: É uma class Bootstrap que adiciona preenchimento e margem a essa tag para melhorar sua aparência;
- **btn-warning**: É uma class Bootstrap que adiciona a cor laranja clara ao fundo e à borda da tag.

Em **style.css**:

```
#start
{
  font-size: 22px;
}
```

P: Como fornecemos estilos à tag HTML com a id “**start**”, e queremos que os mesmos estilos sejam aplicados à tag h3, que acabamos de criar, qual será a melhor opção para alcançarmos esse objetivo?

1. Definir “**#status**” e definir seu estilo de fonte.

2. Adicionar a id “**status**” ao lado da **id “start”** com um separador *dot* (ponto).

R: Opção 2:

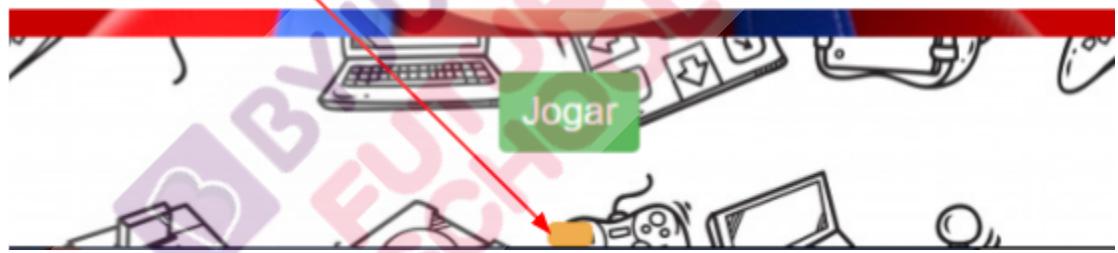
```
#start ,#status
{
  font-size: 22px;
}
```

A partir disso, a tag h3 que criamos terá o mesmo estilo de fonte da id “**start**”.

Saída:

```
<body background="background.jpg">
<center>

    <div class="btn btn-primary heading">
        <h3>IA MARIO</h3>
        <button class="btn btn-primary" data-toggle="modal" data-target="#myModal">Instruções</button>
    </div>
    
    <br><br>
    <button class="btn btn-success" onclick="startGame()" id="start">Jogar</button>
    <br><br>
    <h3 id="status" class="btn btn-warning"></h3>
</center>
</body>
```



- Ainda não fornecemos um texto a essa tag h3, portanto, ela apresenta um espaço vazio.

13. Adicione duas tags br.

```

<br><br>
<button class="btn btn-success" onclick="startGame()" id="start">Jogar</button>
<br><br>
<h3 id="status" class="btn btn-warning"></h3>
<br><br>
```

- Adicionamos essas duas tags br para que haja uma quebra de linha entre a tag h3 e a próxima tag div que conterá o canvas p5.js.

14. Defina uma div vazia para conter o canvas p5.js. O motivo disso será explicado nas próximas aulas. Além disso, forneça a ID **canvas**, pois ela será utilizada para ser consultada a partir do código JS.

```

<br><br>
<button class="btn btn-success" onclick="startGame()" id="start">Jogar</button>
<br><br>
<h3 id="status" class="btn btn-warning"></h3>
<br><br>
<div id="canvas"></div>
```

15. Defina uma div vazia para conter a visualização ao vivo da webcam. O motivo disso será explicado nas próximas aulas. Forneça a ID **gameConsole**, pois a utilizaremos para consultá-la no código JS.

```

<br><br>
<button class="btn btn-success" onclick="startGame()" id="start">Jogar</button>
<br><br>
<h3 id="status" class="btn btn-warning"></h3>
<br><br>
<div id="canvas"></div>
<div id="gameConsole"></div>
```

A saída parecerá incompleta, pois ainda não finalizamos nossos códigos HTML e CSS. Completaremos esses códigos na próxima aula.

OBSERVAÇÃO:

Sempre que o aluno executar o código, **CERTIFIQUE-SE DE QUE ELE O EXECUTE CLICANDO NO BOTÃO GO LIVE DO VISUAL STUDIO. ISSO RESULTARÁ NA EXECUÇÃO DOS ARQUIVOS NO LIVE SERVER DO VISUAL STUDIO**



Visto que utilizamos um arquivo de imagem e áudio, e a biblioteca p5.js não nos permite executar esses tipos de arquivos do sistema local, precisaremos executá-los em um **server**.

A professora encerra o compartilhamento de tela

Agora é a sua vez.

- Peça para o aluno pressionar a tecla ESC para retornar ao painel
- Ajude o aluno a iniciar o compartilhamento de tela
- A professora ativa o modo tela cheia

Passo 3:
Atividade Dirigida pelo Aluno
(10 min)

Você deve fazer o download da pasta **Mario-Game.zip** da [Atividade do Aluno 1](#). **Mario-Game.zip** é uma pasta compactada (.zip). Você precisará descompactá-la para obter a pasta **Mario-Game**. Já mencionei o conteúdo dessa pasta no início da aula.

- Você deve iniciar adicionando os código HTML, na pasta [index.html](#), e CSS, na pasta [style.css](#).

[**Atividade do Aluno 1 - CÓDIGO FONTE**](#)

[**Atividade do Aluno 2 - DIAGRAMA DO CÓDIGO**](#)

A pasta **Mario-Game.zip**, da [Atividade do Aluno 1](#) é uma pasta compactada (.zip). Você deve descompactá-la para obter a pasta **Mario-Game** com os arquivos que vamos trabalhar.

Em seguida, o aluno deve começar adicionando os códigos HTML em [index.html](#) e CSS em [style.css](#).

Após o aluno completar o código, segundo as instruções, peça para que **execute o código**.

CERTIFIQUE-SE DE QUE O ALUNO REALIZE OS TESTES CLICANDO NO BOTÃO GO LIVE DO VISUAL STUDIO. COM ISSO, O CÓDIGO SERÁ EXECUTADO NO LIVE SERVER DO VISUAL STUDIO



Visto que utilizamos arquivos de áudio e imagem, e a biblioteca

		<p>p5.js não nos permite executar qualquer arquivo de áudio ou imagem do sistema local, precisaremos executá-los em um server.</p> <p>Hospedaremos os arquivos do jogo apenas quando completarmos todo o código do projeto. Portanto, mantenha seus arquivos em segurança. Por favor, não modifique os arquivos, pois, como eu disse: “alterar qualquer trecho de código pode corromper o jogo”.</p> <p>Hospedaremos o Jogo Mario com IA apenas quando o código para o jogo estiver completo.</p>
A professora ajuda o aluno a encerrar o compartilhamento de tela		
Step 4: Wrap-Up (3 mins)	<p>Você se saiu muito bem hoje. Ótimo! Você receberá duas tiradas de chapéu.</p> <p>P: Por que utilizamos data-toggle? R: Utilizamos data-toggle para abrir o pop-up. Se data-toggle for adicionado a qualquer elemento HTML, ele resulta na abertura de um pop-up quando o elemento for pressionado.</p> <p>P: Por que utilizamos data-target? Dê um exemplo. R: data-target nos ajuda a abrir o elemento HTML cuja id passamos dentro de data-target. Por exemplo: Se criarmos um botão e, dentro dele, escrevermos o atributo data-target, e dentro de</p>	<p><i>(Dê, no mínimo, 2 tiradas de chapéu)</i></p> <p>Pressione o ícone de Tirada de Chapéu para Resolvendo Atividades Criativamente.</p>  <p>Pressione o ícone de Tirada de Chapéu para Ótima Pergunta.</p>  <p>Pressione o ícone de Tirada de Chapéu para “Você se Concentrou”.</p>  <p>Caso não haja tempo para</p>

	<p>data-target escrevermos a ID HTML “data”, então, quando o botão for pressionado, o elemento HTML com a ID “data” será aberto.</p>	<p>realizar as Atividades Adicionais, peça para o aluno realizá-las após a aula. As Atividades Adicionais estão presentes apenas no painel sob Atividades do Aluno</p> <p>Além disso, lembre o aluno de consultar as Atividades de Referência do Aluno para ampliar seu conhecimento. Isso também deve ser feito após a aula.</p>
--	---	--

Para a solução de todas as Atividades Adicionais, abra a Atividade da Professora 5 e navegue até a aula C137

Atividade Adicional 1:

Execute a [Atividade do Aluno 3](#) do painel

A **ATIVIDADE** e as **DICAS** são mencionadas no próprio website.

Atividade Adicional 2:

Execute a [Atividade do Aluno 4](#) do painel

A **ATIVIDADE** e as **DICAS** são mencionadas no próprio website.

Atividade Adicional 3:

Execute a [Atividade do Aluno 5](#) do painel

A **ATIVIDADE** e as **DICAS** são mencionadas no próprio website.

Atividade Adicional 4:

Execute a [Atividade do Aluno 6](#) do painel

A **ATIVIDADE** e as **DICAS** são mencionadas no próprio website.

Atividade Adicional 5:

Execute a [Atividade do Aluno 7](#) do painel

A **ATIVIDADE** e as **DICAS** são mencionadas no próprio website.

A professora clica em

 **Terminar Aula**

Atividade	Nome da Atividade	Links
Atividade da Professora 1	JOGO MARIO BASE	https://amadolucas.github.io/Mario_Replica/
Atividade da Professora 2	JOGO MARIO COM IA	https://amadolucas.github.io/IA_MARIO/
Atividade da Professora 3	DIAGRAMA DO CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/485c5074-5372-4ad0-a9f2-ff0fefbc558d.pdf
Atividade da Professora 4	CÓDIGO FONTE COMPLETO	https://s3-whjr-curriculum-uploads.whjr.online/8bf56ccc-ece4-4210-aa18-1f091d865de5.zip
Atividade da Professora 5	SOLUÇÃO DAS ATIVIDADES ADICIONAIS	https://docs.google.com/spreadsheets/d/1o03HSCAXB1r8Xhp8s3AWGEIx1y3OFgg1BRq5vCW1fm4/edit#gid=0
Atividade do Aluno 1	CÓDIGO FONTE	https://s3-whjr-curriculum-uploads.whjr.online/59a0149e-50cf-4808-aa3c-27c23ab5beb9.zip
Atividade do Aluno 2	DIAGRAMA DO CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/485c5074-5372-4ad0-a9f2-ff0fefbc558d.pdf
Atividade de Referência do Aluno 1	REFERÊNCIA PROPRIEDADE CSS DISPLAY	https://www.w3schools.com/cssref/pr_class_display.asp
Atividade de Referência do Aluno 2	REFERÊNCIA PROPRIEDADE CSS FLOAT	https://www.w3schools.com/css/css_float.asp
Solução do Projeto	IA PING PONG - PARTE 1	https://s3-whjr-curriculum-uploads.whjr.online/4ef2106e-f221-41af-bcb8-9e2e2f9dd00f.pdf