
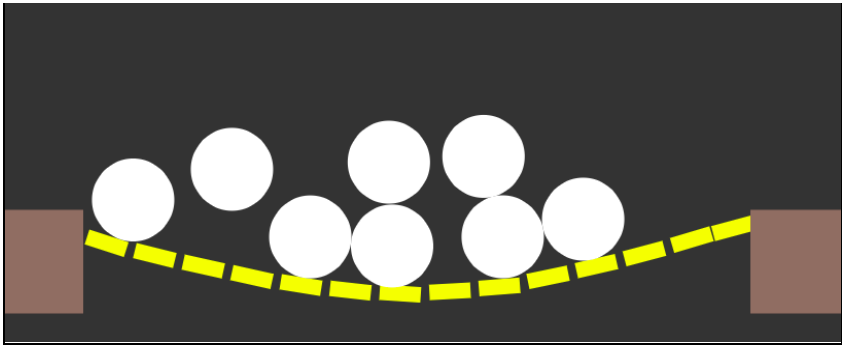



Tópico	PARTINDO A CORDA	
Descrição da Aula	O aluno aprende como criar uma função para soltar a fruta da corda. O aluno criará um sprite de coelho e adicionará uma imagem de plano de fundo.	
Aula	C30	
Duração da aula	45 min	
Objetivo	<ul style="list-style-type: none"> • Solte a fruta ao cortar a corda. • Crie o corpo para o coelho. • Adicione uma imagem para o plano de fundo e para o coelho. 	
Recursos Necessários	<ul style="list-style-type: none"> • Recursos da Professora: <ul style="list-style-type: none"> ○ Laptop com conectividade à internet ○ Editor VS Code ○ Fone de ouvido com microfone ○ Bloco de notas e caneta • Recursos do Aluno: <ul style="list-style-type: none"> ○ Laptop com conectividade à internet ○ Editor VS Code ○ Fone de ouvido com microfone ○ Bloco de notas e caneta 	
Estrutura da aula	Introdução - Opção de Apresentação de Slides Atividade Dirigida pela Professora Atividade Dirigida pelo Aluno Fechamento - Opção de Apresentação de Slides	10 min 10 min 20 min 5 min
SESSÃO DE INTRODUÇÃO - 10 min		
<div>  </div> <p>A professora inicia a apresentação de slides do slide 1 ao 11. Consulte as anotações da oradora e siga as instruções em cada slide.</p>		

Detalhes da atividade	Solução/Orientações
<p><i>Olá <nome do aluno>. Como você está hoje? Você está empolgado pela aula de hoje?</i></p> <p>Execute a apresentação do slide 1 ao slide 3.</p> <p>A seguir estão os resultados esperados da sessão de introdução:</p> <ul style="list-style-type: none"> Cumprimente o aluno. Revise as atividades da última aula. Questionários. 	<p>Resposta: Oi, obrigado. Sim, eu estou empolgado!</p> <p>Clique na aba de apresentação de slides e apresente os slides.</p>
Sessão de Perguntas e Respostas	
Pergunta	Resposta
<p>Selecione a opção correta para criar uma bridge (ponte) usando uma classe Bridge e criar um jointPoint usando a Classe Base.</p> <p>A. <pre>/*bridge = new Bridge(15, { x: width / 2 - 400, y: height / 2 }); jointPoint = new Base(width - 600, height / 2 + 10, 40, 20, "#8d6e63", true);*/</pre></p> <p>B. <pre>/*bridge = new Base(15, { x: width / 2 - 400, y: height / 2 }); jointPoint = new Base(width - 600, height / 2 + 10, 40, 20, "#8d6e63", true);*/</pre></p> <p>C. <pre>/*bridge = new Base(15, { x: width / 2 - 400, y: height / 2 }); jointPoint = new Bridge(width - 600, height / 2 + 10, 40, 20, "#8d6e63", true);*/</pre></p> <p>D. <pre>/*bridge = new Bridge(15, { x: width / 2 - 400, y: height / 2 }); jointPoint = new Bridge(width - 600, height / 2 + 10, 40, 20, "#8d6e63", true);*/</pre></p>	A
<p>Selecione a opção correta para passar bridge.body e jointPoint em Matter.Composite.add().</p>	A

 <p>A. Matter.Composite.add(bridge.body, jointPoint); B. Matter.Composite.add(jointPoint); C. Matter.Composite.add(jointPoint, bridge.body); D. Matter.Composite.add(bridge.body);</p>	
Continue a sessão de introdução	
Detalhes da atividade	Solução/Orientações
<p>Execute a apresentação do slide 4 ao slide 11 para definir a declaração do problema.</p> <p>A seguir estão os resultados esperados da sessão de introdução:</p> <ul style="list-style-type: none"> Revise como carregar imagens 	<p>Narre os slides usando gestos com as mãos e métodos de modulação de voz para atrair mais interesse dos alunos.</p>
<p>Professora encerra a apresentação de slides</p> 	
ATIVIDADE DIRIGIDA PELA PROFESSORA - 10 min	
Professora Inicia o Compartilhamento de Tela	
<p><u>DESAFIO</u></p> <ul style="list-style-type: none"> Adicionar a imagem do plano de fundo e da fruta. 	
Ação da Professora	Ação do Aluno

Atividade Dirigida pela Professora 1

Na aula anterior, criamos a nossa corda e penduramos nela o corpo da fruta.

Nesta aula, vamos escrever o código para remover a fruta da corda quando o usuário pressionar o botão “cortar” e, assim, fazer a fruta cair.

Também criaremos o corpo do coelho e adicionaremos imagens para ele e para o fundo.

Primeiro vamos adicionar todas as imagens. Precisamos de imagens para o plano de fundo, para o coelho e para o botão cortar.

As imagens estão presentes na pasta.

Agora precisamos carregar as imagens no nosso código.

Você pode me dizer como vamos fazer isso?

Muito bom, vamos escrever a função **preload()**.

Vamos escrever este código no arquivo **sketch.js**.

Carregamos os ativos na função **preload()** porque queremos carregar todas as imagens antes do nosso código principal começar. A função **preload()** é executada primeiro, portanto todas as outras funções e instruções são executadas após a função **preload**.

Primeiro precisamos declarar 3 variáveis para as imagens, tais como:

var bg_img, fruta_img, bunny_img;

Definimos a função **preload()** e agora adicionaremos as imagens do fundo, da fruta e do coelho.

Carregue a imagem usando a função **loadImage()** e passe o caminho da imagem como parâmetro dentro da função **loadImage()**.

A professora faz o download e abre o código da [Atividade da Professora 1](#) no editor VS Code.

Resposta:

Usando **loadImage()** e a função **preload()**.

Desta forma, as imagens serão carregadas e iremos atribuí-las às variáveis que criamos anteriormente.

```
function preload()
{
  bg_img = loadImage('background.png');
  food = loadImage('melon.png');
  rabbit = loadImage('Rabbit-01.png');
}
```

Agora que temos as imagens, podemos colocá-las em seus respectivos lugares.

Mas, antes de fazer isso, precisamos adicionar mais uma instrução na função **setup()**.

Que é definir o modo de imagem como no centro. Com isso, ao exibir a imagem na tela, teremos o ponto central das imagens.

Por padrão, quando criamos a imagem na tela, ele toma o canto superior esquerdo como ponto de origem, mas aqui estamos mudando para o centro. Então, agora, quando desenharmos a imagem na tela, especificaremos sua posição central como as coordenadas x e y.

Para fazer isso, precisamos escrever a função **imageMode()** e adicionar **CENTER** como parâmetro na função.

Agora carregamos as imagens e vamos exibi-las na tela.

A primeira é a imagem do plano de fundo.

Usaremos a função **image()** para exibir a imagem na tela. Esta função tomará como argumentos a posição x, y da imagem e a sua largura e altura.

A imagem é armazenada na variável **bg_img**, definiremos a posição x e y como largura/2 e altura/2 porque definimos

o **imageMode** como **CENTER**. Em seguida, seguido pelo tamanho da imagem como o tamanho da tela, que é **500** e **700**, respectivamente.

Na função **draw()**, escreva a função **image()** e passe os argumentos correspondentes.

```
function setup()
{
  createCanvas(500,700);
  frameRate(80);
  engine = Engine.create();
  world = engine.world;
  ground = new Ground(200,680,600,20);

  rope = new Rope(7,{x:245,y:30});
  fruit = Bodies.circle(300,300,20);
  Matter.Composite.add(rope.body,fruit);

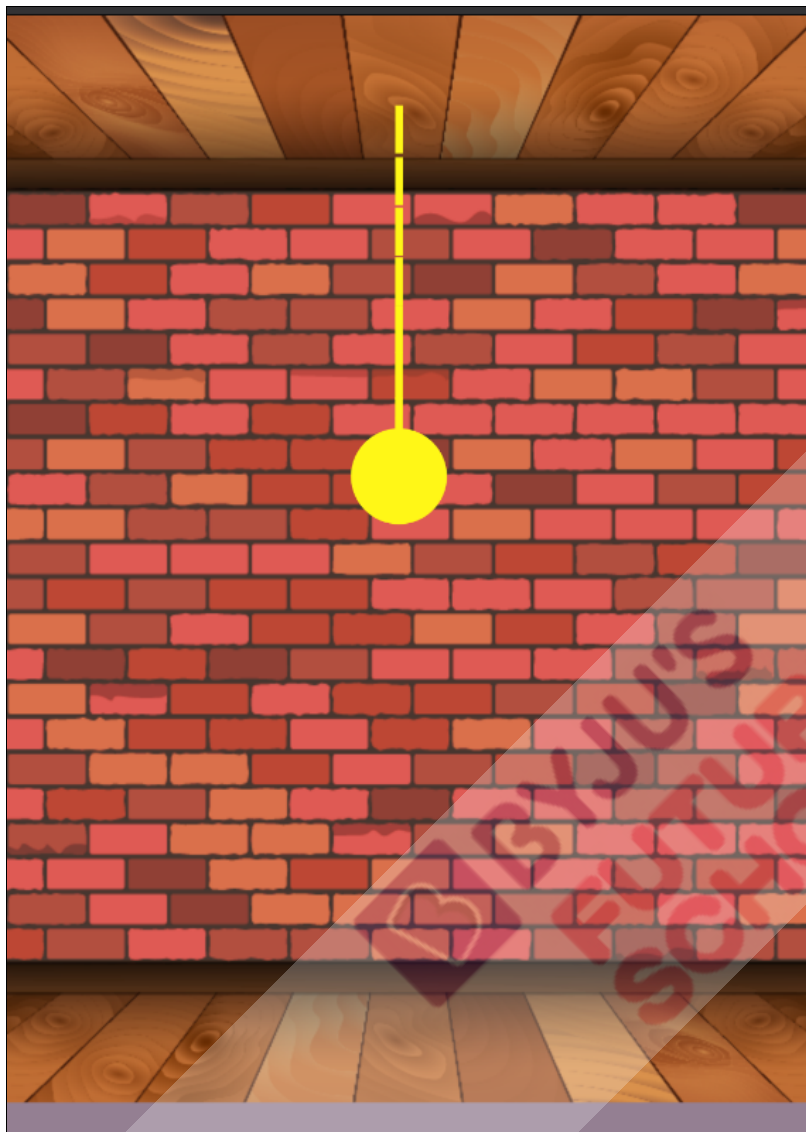
  fruit_con = new Link(rope,fruit);

  rectMode(CENTER);
  ellipseMode(RADIUS);
  textSize(50);

  imageMode(CENTER);
}
```

A professora executa o código.

Quando você executa, clicando no botão **GoLive**, o código irá mostrar o plano de fundo do nosso jogo.



Agora vamos adicionar a imagem da fruta.

Já criamos o corpo da fruta e agora estamos criando um círculo usando as posições do corpo da fruta.

Vamos adicionar a imagem da melancia no mesmo local, mas precisamos remover o círculo.

Também faremos com que a imagem da fruta pareça um pouco maior e real, portanto, manteremos a largura e a altura da imagem da melancia em **60, 60**.

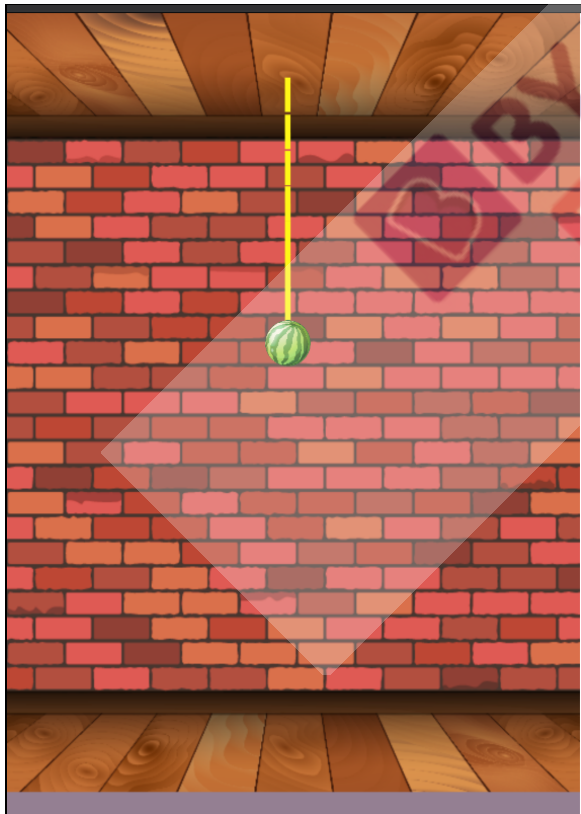
Aqui, **Engine.update(engine)** atualizará o resultado.

```
function draw()
{
  background(51);
  image(bg_img,width/2,height/2,500,700);
  ground.show();
  rope.show();

  image(fruit_img,fruit.position.x,fruit.position.y,60,60);


  Engine.update(engine);
}
```


Resultado:



Nós adicionamos o plano de fundo e a imagem da fruta.

<p>Agora é sua vez de adicionar um botão e funções para cortar a corda e soltar a fruta.</p> <p>O que você acha? Como podemos criar um botão na tela?</p> <p>Ok, vamos descobrir.</p> <p>Por favor, compartilhe sua tela comigo.</p>	<p>REA: Variada.</p>
--	---------------------------------

<p>A professora inicia a apresentação de slides  : slide 12 ao slide 18.</p>	
<p>Apresente o slide para definir o contexto da atividade do aluno.</p>	
<ul style="list-style-type: none"> • Crie uma função para adicionar um botão e uma função para soltar a fruta. • Adicione o sprite do coelhinho. 	

<p>A professora encerra a apresentação de slides </p>	
<p>A Professora Para o Compartilhamento de Tela</p>	
<p>ATIVIDADE DIRIGIDA PELO ALUNO - 20 min</p>	
<ul style="list-style-type: none"> • Peça ao aluno para pressionar a tecla ESC para voltar ao painel. • Oriente o aluno a iniciar o Compartilhamento de Tela. • A professora entra em Tela Cheia. 	
<p><u>ATIVIDADE</u></p> <ul style="list-style-type: none"> • Crie o botão e a função para soltar a fruta. • Adicione o sprite do coelho. 	
<p>Ação da Professora</p>	<p>Ação do Aluno</p>
<p>Atividade Dirigida pelo Aluno-1</p>	

Adicionamos as imagens da fruta e do plano de fundo.

Agora vamos criar o Sprite para o coelhinho.

Fruta, solo e corda são corpos físicos que estão usando a biblioteca de física **matter.js**.

Não criaremos um corpo físico para o coelhinho pois só precisamos dele para reproduzir sua animação e detectar a colisão com as frutas.

É por isso que vamos usar a **biblioteca p5.play**, que usamos para fazer o jogo T-rex.

A biblioteca já foi adicionada ao código que você baixou agora, então podemos prosseguir diretamente para a criação do sprite.

Para criar um sprite, primeiro, declare uma variável **var bunny**;

Estamos escrevendo este código no arquivo **sketch.js**.

Agora, na função **setup()**, criamos o sprite usando a função **createSprite()**.

Nesta função, precisamos passar as posições x, y, e a largura e altura do sprite.

Também precisamos adicionar a imagem ao coelhinho e, para isso, usaremos a função **bunny.addImage()**.

Também definiremos a imagem para o sprite do coelhinho e definiremos a escala do sprite como **0.2**, porque a imagem que temos é muito maior e pode não caber em nossa tela, então precisamos reduzi-la.

O aluno faz o download do código [Atividade do Aluno 1](#) e abre no editor VS Code.

```
function setup() {  
  createCanvas(500,700);  
  frameRate(80);  
  engine = Engine.create();  
  world = engine.world;  
  bunny = createSprite(250,650,100,100);  
  bunny.addImage(bunny_img);  
  bunny.scale = 0.2;  
}
```

Se você executar o código, não conseguirá ver o coelho na tela.

Pode me dizer por quê?

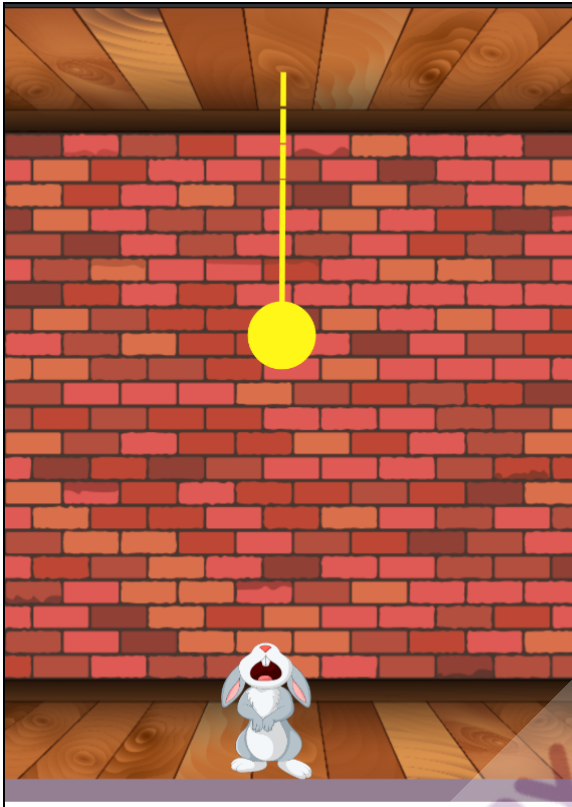
Vamos adicionar **drawSprite()** dentro da função **draw()** e executar nosso código.

REA:

Nós precisamos desenhar o sprite usando a função **drawSprite()**.

```
function draw()  
{  
  background(51);  
  image(bg_img,width/2,height/2,500,700);  
  ground.show();  
  rope.show();  
  
  image(fruit_img,fruit.position.x,fruit.position.y,60,60);  
  Engine.update(engine);  
  drawSprites();  
}
```

Resultado:



Agora precisamos criar uma função para soltar a melancia e quebrar a corda.

Temos duas coisas para realizar:

- Cortar a corda do ponto superior onde ela está conectada.
- Cortar a fruta da corda.

Para cortar a corda do topo, vamos chamar uma função da classe rope, que é chamada **break()**. Isso vai soltar a corda e removê-la de cena.

Em seguida, para separar a fruta da corda, precisamos excluir a restrição entre a corda e a fruta, removendo essa restrição do mundo.

Vamos criar esta função na classe **Link** para que possamos chamá-la usando o objeto de ligação.

Para isso, usaremos a função **World.remove()**, que removerá a restrição do mundo.

Para chamar a função **detach()** & **break()**, criaremos uma nova função no arquivo **sketch.js** chamada **drop()** que chamaremos (executar) pressionando um botão.

Também tornaremos a restrição de frutas **null**. Para que não afete a fruta.

```
class Link{
  constructor(bodyA,bodyB)
  {
    var lastlink = bodyA.body.bodies.length-2;
    this.link = Constraint.create(
      {
        bodyA:bodyA.body.bodies[lastlink],
        pointA:{x:0,y:0},
        bodyB:bodyB,
        pointB:{x:0,y:0},
        length:-10,
        stiffness:0.01
      });
    World.add(engine.world,this.link);
  }

  detach()
  {
    World.remove(engine.world,this.link);
  }
}
```

Agora vamos definir a função **drop()**, onde vamos partir a corda usando a função **rope.break()**, e removeremos a restrição da fruta usando a função **fruit_con.detach()**,

tornando **fruit_con** como null.

```
function drop()
{
  rope.break();
  fruit_con.detach();
  fruit_con = null;
}
```

Criamos nossa função, agora vamos criar um botão e adicionar essa função com esse botão para que possamos soltar nossa fruta.

Em **p5.js**, temos funções embutidas para criar botões na tela, e já fizemos esse exercício antes em várias aulas antes.

Onde usamos a função **createButton()**. Mas isso cria um botão simples na tela e ele não parece ser muito atraente visualmente. Portanto, agora usaremos uma função diferente chamada **createImg()**.

Isso pega uma imagem como parâmetro e faz com que ela funcione como um botão.

Primeiro, defina uma variável como um botão **var**.

Então, na função **setup()**, crie o botão de imagem usando a função **createImg()** e passe a imagem que queremos mostrar neste botão.

Também precisamos especificar o tamanho e a posição do botão.

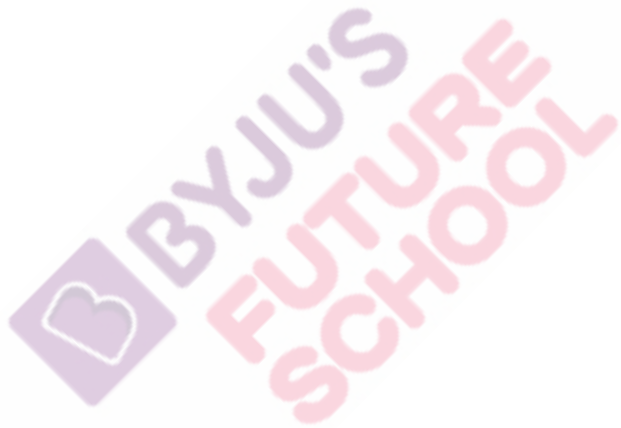
Finalmente, vamos adicionar a função com a ajuda da função **mouseClicked()**.

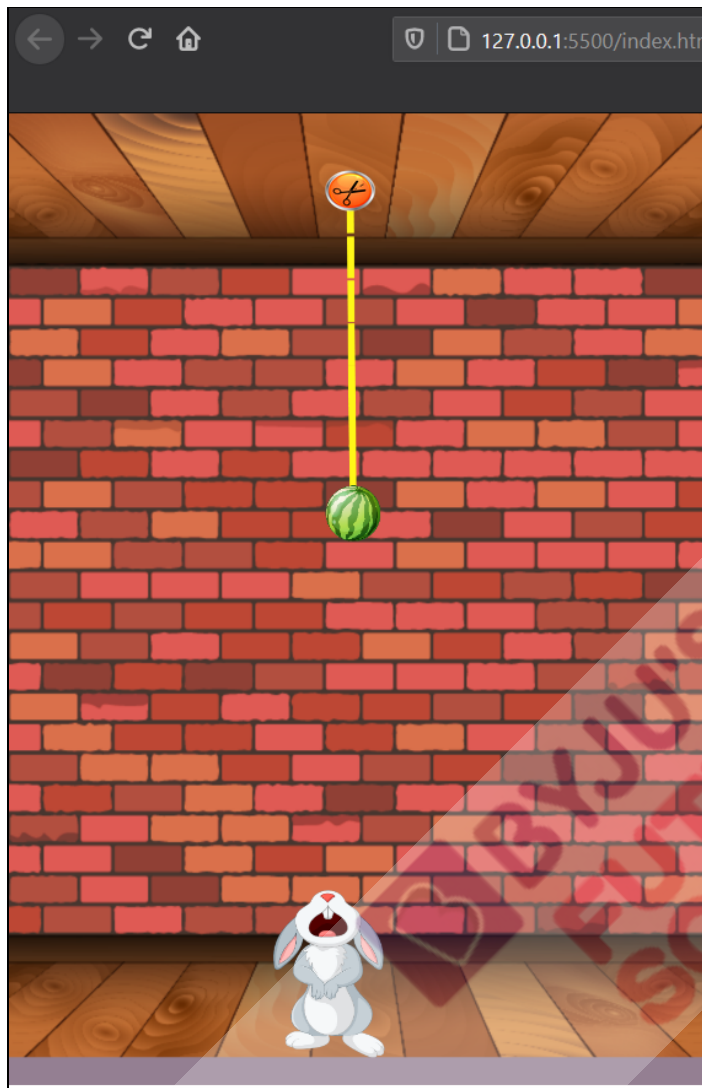
Na função **mouseClicked()**, passamos a função que queremos executar ao clicar no botão.

Vamos executar o código e ver se está funcionando.

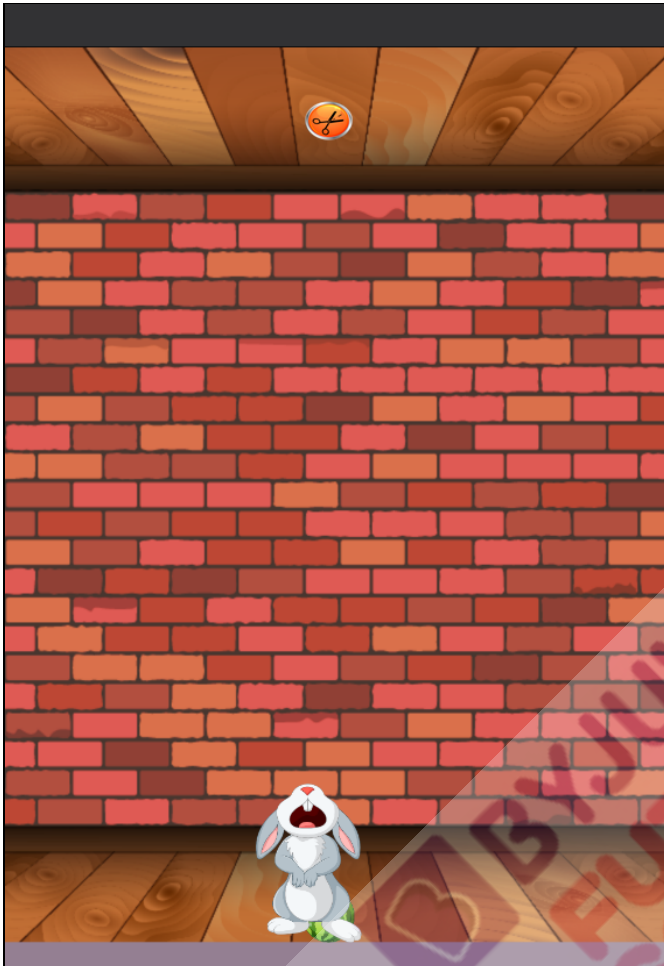
```
button = createImg('cut_btn.png');  
button.position(220,30);  
button.size(50,50);  
button.mouseClicked(drop);
```

Resultado:





Temos o nosso botão, vamos clicar no botão e ver se a nossa fruta cai ou não.



Isso é ótimo, nosso código está funcionando e a fruta cai corretamente.


Você pode tentar algumas coisas. Atualize a página de resultado e clique no botão quando a corda estiver balançando, você verá que a fruta cai em locais diferentes.


Conseguimos completar a estrutura básica do jogo.




Na próxima aula, vamos adicionar animação ao nosso coelhinho e vamos detectar a colisão da fruta com o coelhinho para reproduzir diferentes animações.

A Professora Orienta o Aluno a Parar o Compartilhamento de Tela

SESSÃO DE FECHAMENTO - 5 min

A professora começa a apresentação de slides  dos slides 19 a 28.

Detalhes da atividade	Solução/Orientações
<p>Exiba a apresentação do slide 19 ao slide 28.</p> <p>A seguir estão os resultados esperados da sessão de fechamento:</p> <ul style="list-style-type: none"> • Explique os fatos e curiosidades. • Desafio da próxima aula. • Projeto do dia. • Atividade adicional. 	<p>Oriente o aluno a desenvolver o projeto e compartilhá-lo conosco.</p>
<p>A professora encerra a apresentação de slides </p>	
Sessão de Perguntas e Respostas – Clique no questionário de aula	
Pergunta	Resposta
<p>Por que usamos o comando <code>imageMode(CENTER)</code>?</p> <p>A. Porque, por padrão, quando criamos a imagem na tela, ela assume o canto superior esquerdo como ponto de origem.</p> <p>B. Para desenhar a imagem no centro da tela.</p> <p>C. Para desenhar a imagem no canto superior esquerdo da tela.</p> <p>D. Nenhuma das alternativas acima.</p>	A
<p>Qual dos seguintes comandos é usado para remover o corpo do mundo?</p>	C

<p>A. World.remove(body,world) B. World.remove(world,body) C. World.remove(body) D. body.remove(world)</p>	
<p>Qual função embutida do p5.js é usada para criar botões na tela?</p> <p>A. create.button() B. createButton() C. button.create() D. Createbutton()</p>	<p>B</p>
<p>Encerrar o painel do questionário</p>	
	<p>Você recebeu uma Tirada de Chapéu</p> <p><i>Certifique-se de ter dado pelo menos 2 chapéus durante a aula para:</i></p> <div data-bbox="1019 909 1312 1066">  <p>Resolvendo Atividades Criativamente +10</p> </div> <div data-bbox="1019 1077 1312 1224">  <p>Ótima Pergunta +10</p> </div> <div data-bbox="1019 1234 1312 1381">  <p>Você se Concentrou +10</p> </div>
<p>* Este projeto levará apenas 30 minutos para ser concluído. Motive o aluno a tentar concluí-lo imediatamente após a aula. Panorama do Projeto</p> <p>ESMAGUE OS ZUMBIS - 2</p> <p>Objetivo do Projeto:</p>	<p><i>Os alunos se engajam no projeto com a professora.</i></p>

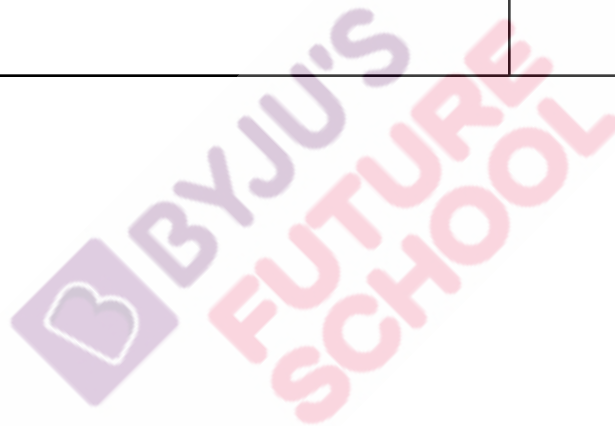
Neste projeto, você usará conceitos semelhantes para criar um zumbi. Adicione uma imagem para o zumbi, as pedras, a ponte e o plano de fundo. Você também pode adicionar um botão que, quando pressionado, fará a ponte desmoronar, liberando as pedras no zumbi.

História:

Uma aldeia distante é sempre perturbada por um zumbi. A única maneira de matar um zumbi é jogando uma pedra em sua cabeça. Você percebeu que o zumbi passa por baixo da ponte para chegar à aldeia. Então você deve planejar como empilhar pedras para jogá-las no zumbi quando ele passar por baixo da ponte.

Eu estou muito animada para ver sua solução para o projeto e sei que você se sairá muito bem.

Até breve!



Links:

Nome da Atividade	Descrição	Link
Atividade da Professora 1	Código modelo	https://github.com/c-amilarorigues/C30-AP1
Atividade do Aluno 1	Código modelo	https://github.com/c-amilarorigues/C30-AA1
Código de Referência	Código de Referência Completo	https://github.com/c-amilarorigues/C30-completo
Solução do Projeto	Esmague os Zumbis 2	https://github.com/ByjusBrazil/PRO_1-1_C30_SolucaoDoProjeto
Link do recurso visual de referência da professora	Recurso visual	https://s3-whjr-curriculum-uploads.whjr.online/ddeb26fc-ef18-4c69-b785-2d0984c30433.html
Referência da Professora do Questionário de aula	Questionário de aula	https://s3-whjr-curriculum-uploads.whjr.online/76fadde3-a001-4ad2-88ef-65f814ba73d1.pdf

Referência da Professora - NÃO SE DEVE ADICIONAR AO PAINEL DE LINKS DE ATIVIDADES

Link do Recurso Visual para Referência da Professora	Recurso visual	https://s3-whjr-curriculum-uploads.whjr.online/3928f66a-5934-4747-91cb-de0b4e801ba.html
--	----------------	---