

Tópico	FRUTA PENDURADA NA CORDA	
Descrição da Aula	O aluno aprenderá a criar um corpo de corda usando a biblioteca matter.js. O aluno também irá prender uma fruta a uma corda usando restrições.	
Aula	C29	
Duração da aula	45 min	
Objetivo	<ul style="list-style-type: none"> • Criar o corpo da corda. • Prenda a fruta com a corda. 	
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 Atividade Dirigida pela Professora Atividade Dirigida pelo Aluno FECHAMENTO	10 min 10 min 20 min 5 min
A Professora inicia a Apresentação Sob os Holofotes		
<p>No início desta aula, você fará a sessão Sob os Holofotes com os pais. Abra o link do aluno (Recursos Visuais) específico desta sessão. Este link não estará disponível na plataforma do aluno: você compartilha a sua tela. O link da professora contém os Speaker Notes (comentários) para que você possa estudá-los e não deve ser compartilhado com o aluno.</p> <p>ATENÇÃO: se a sessão Sob os Holofotes for remarcada, salve o link pois ele não parecerá nos currículos das próximas aulas.</p>		



A professora inicia a apresentação de slides do slide 1 ao 9

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

Ação da Professora	Ação do Aluno
<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. A seguir estão os resultados esperados da sessão de introdução:</p> <ul style="list-style-type: none"> • Cumprimente o aluno. • Revise a atividade da última aula. • Questionários. 	<p>Resposta: Oi, obrigado. Sim, eu estou empolgado!</p> <p>Clique na aba de apresentação de slides e apresente-os.</p>
Sessão de Perguntas e Respostas	
Pergunta	Resposta
<p>Selecione a opção correta para escrever uma condição que verificará se as flechas colidem com os alvos e, em seguida, aumentará a pontuação em 5.</p> <p>A. <pre>/*if (board1Collision board2Collision) { score += 5; }*/</pre></p> <p>B. <pre>/*if (board1Collision.collided && board2Collision.collided) { score += 5; }*/</pre></p> <p>C. <pre>/*if (board1Collision.collided board2Collision.collided) { score += 5; }*/</pre></p> <p>D. <pre>/*if (board1Collision.collided board2Collision.collided) { score = 5; }*/</pre></p>	<p>C</p>

Selecione a opção correta para chamar a função **gameOver()** se **numberOfArrows** for igual a zero.



```
/*if (numberOfArrows == 5) {
|   gameOver();
| }*/
```

A.

```
/*if (numberOfArrows == 0) {
|   gameOver();
| }*/
```

B.

```
/*if (numberOfArrows = 0) {
|   gameOver();
| }*/
```

C.

```
/*if (numberOfArrows == 0) {
|   gameOver;
| }*/
```


D.

B

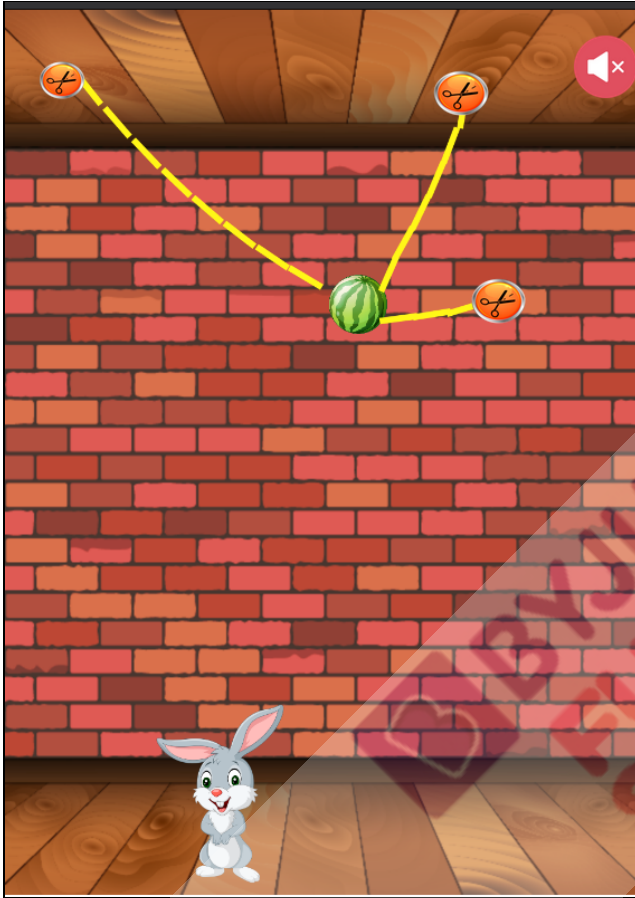
Continue a sessão de introdução

Detalhes da atividade

Solução/Orientações

<p>Execute a apresentação do slide 4 ao slide 9 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"> Elogie o aluno pelo seu desempenho nos questionários. Crie uma classe Ground. 	<p>Narre os slides usando gestos com as mãos e métodos de modulação de voz para atrair mais interesse dos alunos.</p>
<p style="text-align: center;">Professora encerra a apresentação de slides</p> 	
<p style="text-align: center;">ATIVIDADE DIRIGIDA PELA PROFESSORA - 10 min</p>	
<p style="text-align: center;">Professora Inicia o Compartilhamento de Tela</p>	
<p style="text-align: center;"><u>DESAFIO</u></p> <ul style="list-style-type: none"> Criar uma classe Ground. 	
Ação da Professora	Ação do Aluno
<p>Atividade Dirigida pela Professora 1:</p> <p><i>A professora faz o download da Atividade da Professora 1 e executa no editor VS Code.</i></p> <p>Nas últimas aulas, nós começamos a usar a biblioteca física matter.js e construímos uma Invasão Pirata com ela.</p>	
<p>A partir desta aula começaremos com um novo jogo chamado Alimente o Coelho!</p> <p>Antes de começarmos com o código, vamos dar uma visão geral rápida do jogo.</p> <p><i>Observação: Imagem a ser apresentada no Auxílio Visual.</i></p> <p>Neste jogo, temos uma melancia pendurada na corda. O coelhinho está esperando no chão para comer a fruta. Quando o usuário clica no botão para cortar a corda, a melancia cai e o coelho a come.</p>	

Como vimos no auxílio visual, teremos diferentes arranjos de cordas nas próximas aulas, de forma que o usuário tenha que escolher qual corda cortar para a melancia cair no coelho.



Então, vamos começar com este jogo:

Nosso coelhinho estará de pé no chão.

A primeira coisa que precisamos fazer é criar o corpo do solo e, para isso, criaremos a classe Ground.

- Crie um novo arquivo e nomeie-o **ground.js**.
- Agora adicione este arquivo ao **index.html** para que possamos usá-lo em nosso código.

```
<body>  
  <script src="matter.min.js"></script>  
  <script src="p5.play.js"></script>  
  <script src="rope.js"></script>  
  <script src="ground.js"></script>  
  <script src="sketch.js"></script>  
  
</body></html>
```

Abra o arquivo **ground.js**, aqui vamos criar a classe **Ground**.

O solo (ground) será um corpo estacionário, portanto, precisamos apenas criar a função **constructor()** e a função para exibi-lo.

No construtor, especificaremos as posições **x**, **y** e a **width** (largura) e **height** (altura), que serão inseridas pelo usuário ao criar o objeto de **solo** para a classe **ground()**.

*Observação: **Flags** (sinalizadores) são variáveis que podemos usar para alterar o estado de um objeto, os sinalizadores têm apenas valores booleanos **true** (verdadeiro) ou **false** (falso).*

Para as **options** (opções), estamos apenas mantendo a sinalização **isStatic** como **true** porque não queremos que o solo se mova.

```
class Ground
{
  constructor(x, y, w,h)
  {
    let options = {
      isStatic:true
    };

    this.body = Bodies.rectangle(x, y, w, h, options);
    this.w = w;
    this.h = h;
    World.add(world, this.body);
  }
}
```

O solo é um corpo retangular, por isso estamos usando a função **Bodies.rectangle()** para criar este corpo.

Assim que a criação do corpo estiver concluída, adicionaremos este corpo ao mundo.

A função **constructor()** só nos permitirá criar o corpo mas, se quisermos mostrar o solo no jogo, precisamos escrever uma função para exibi-lo.

Crie um novo método **show()** dentro da classe **Ground**.

Para desenhar um corpo, precisamos de suas **posições x e y**.

Você pode me dizer como podemos obter as posições x e y do corpo na aula?

Excelente! Armazenamos o valor da posição do solo em uma variável, ou seja, em **pos**.

REA:
Usando **this.body.position**.


```
show() {
  let pos = this.body.position;
  push();
  rectMode(CENTER);
  noStroke();
  fill(148,127,146);
  rect(pos.x,pos.y, this.w, this.h);
  pop();
}
```

Quando criamos um retângulo em p5, ele assume o ponto superior esquerdo como a origem do retângulo. Mas para alterar o ponto de origem do retângulo do canto superior esquerdo para o centro, usamos a função **rectMode()**. Nesta função, passamos **CENTER** como um argumento.

Isso criará um retângulo com o ponto central como origem. Como queremos ter uma cor e um traço específicos para o solo, é por isso que estamos usando a função **push()**, e depois adicionaremos as propriedades de cor e traço do solo. **noStroke()** serve para não ter nenhum contorno ao redor do corpo.

Usamos a função **fill()** para adicionar cor a uma **forma**. O código **RGB** da cor é passado para a função **fill()** como um argumento.

Em seguida, criamos a forma do solo, que é um retângulo, usando a função **rect()**.

E então escrevemos a função **pop()**. Isso vai encerrar nossa criação de formas.

Nossa classe **Gound** está completa, agora podemos criar o objeto da classe.

Você pode me dizer onde vamos fazer isso?

Muito bem!

REA:
Sim! No arquivo **sketch.js**.

Primeiro, declare uma variável como **ground**. Então, na função **setup()**, crie o objeto da classe **ground** e armazene-o nesta variável.

Você sabe como criar o objeto da classe?

Ótimo, vamos escrever.

Ao criar um objeto de **solo**, precisamos fornecer suas posições x, y, width (largura) e height (altura).

Nosso jogo estará no modo retrato. É por isso que o tamanho da nossa tela é **(500,700)**.

Precisamos ter isso em mente ao definir a posição e a largura do solo.

Posição x é **200**, y é **690**, a largura do corpo do solo é **200** e a altura é **20**.

REA:

Usando a palavra-chave **new** e o **nome da classe**.

```
function setup() {
  createCanvas(500,700);
  frameRate(80);
  engine = Engine.create();
  world = engine.world;

  ground = new Ground(200,690,600,20);
}
```

Agora a nossa criação de objeto está completa.

Agora podemos mostrar o solo.

Como podemos exibir o solo?

Ótimo, fazemos isso na função **draw()**.

Agora podemos executar nosso código.

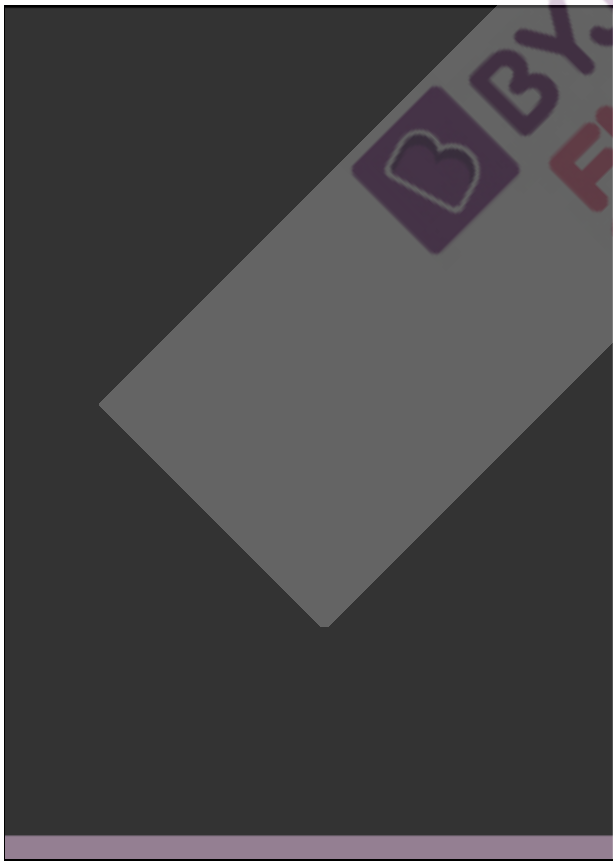
REA:

Chamando a função **ground.show()**.


Clique no botão **Go Live** para abrir uma janela do navegador onde podemos ver nossa tela e o solo.


```
function draw()  
{  
  background(51);  
  ground.show();  
  
  Engine.update(engine);  
}
```

Resultado:



<p>A primeira etapa está concluída. Aqui, criamos uma classe ground e um objeto de solo.</p> <p>Agora podemos ver o corpo do solo na tela.</p> <p>Na próxima etapa, você precisa criar o corpo da corda e da fruta.</p> <p>Agora é sua vez de escrever o código para adicionar a corda e a fruta.</p>	

Professora inicia a apresentação de slides  : slides 10 a 18	
<p>Execute o slide da apresentação para definir o contexto da atividade do aluno.</p> <ul style="list-style-type: none"> • Criar um objeto da classe Rope (corda). • Explicar o módulo Composto. • Pendurar frutas na corda. 	

Professora encerra a apresentação de slides 	
Professora Para o Compartilhamento de Tela	
ATIVIDADE DIRIGIDA PELO ALUNO - 20 min	
<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 align="center"><u>ATIVIDADE</u></p> <ul style="list-style-type: none"> • Crie uma corrente e pendure-a na tela. • Crie o corpo da fruta e prenda-o em uma corda. 	
Ação da Professora	Ação do Aluno

Passo: 2 - Atividade Dirigida pelo Aluno

Agora precisamos criar o objeto corda e pendurá-lo em um ponto na tela, então vamos criar um corpo de fruta e prendê-lo na corda.

Também adicionaremos imagens de plano de fundo e imagens de fruta para tornar nosso jogo visualmente atraente.

Você pode perceber que já temos um arquivo chamado **rope.js** que contém o código para criar uma corda.

Observando o código do **rope.js**:

- Em **constructor()**, tomamos dois parâmetros:
nlinks - número de links e
pointA - pontos de conexão.

Usando a função **Composites.stack()**, criamos os vários corpos retangulares e os armazenamos na variável **rect**.

E então, usando o **Constraints.create()**, adicionamos as restrições à cadeia que conecta todos os corpos da cadeia como se tivéssemos um cordão em um colar.

Temos a função **break()** que nos ajuda a quebrar a cadeia. Ela simplesmente torna o corpo da corda nulo.

A professora pode checar os detalhes de composições em <https://brm.io/matter-js/docs/classes/Composites.html>

Para o nosso código, vamos usar o **rope.js** apenas para criar a função **rope** (corda) e **break()** para quebrar a corda quando o usuário clicar no botão cortar (que será adicionado nas próximas aulas).

Adicione **rope.js** ao **index.html**

Para criar uma corda, precisamos de duas informações importantes: a primeira é o comprimento da corda e onde queremos pendurá-la.

Para definir o comprimento da corda, definimos um número que cria essas muitas seções na corda. Uma

REA:

O aluno faz o download do código da Atividade do Aluno 1 e o executa no editor VS Code.

*O aluno adiciona **rope.js** ao **index.html**.*

seção é essencialmente um retângulo, então você pode imaginar nossa corda como vários retângulos conectados entre si.

Em seguida, precisamos do ponto onde vamos pendurar a corda, que será uma certa posição x e y na tela.

Primeiro, você define uma variável chamada **rope**.

Então, na função **setup()**, crie um objeto da classe **Rope**.

Ao criar o objeto, primeiro escrevemos o comprimento da corda, que definiremos como **6** e, em seguida, as posições x e y, nesse caso x seria **245** e y seria **30**.

```

1  let engine;
2  let world;
3  var ground;
4  var rope;
5
6  function setup() {
7    createCanvas(500,700);
8    frameRate(80);
9    engine = Engine.create();
10   world = engine.world;
11   ground = new Ground(200,690,600,20);
12
13   rope = new Rope(6,{x:245,y:30});
14 }
```

A próxima etapa é chamar a função **rope.show()** na função **draw()** para que possamos ver a corda pendurada.

Agora vem a parte divertida de criar o corpo da fruta e pendurá-lo com a nossa corda.

Alguma ideia de como podemos fazer isso?

REA:
Respostas variadas.

Primeiro, vamos criar um corpo de fruta.

Declare uma variável como **var fruit;**

Em seguida, na função setup, primeiro criamos **fruit_options** para o corpo de fruta. Nas opções, estamos apenas definindo a densidade da fruta como **0.001**.

Em seguida, criamos um corpo de fruta usando a função **Bodies.circle()**.

Na função, precisamos fornecer as posições x e y, e o raio da fruta junto com **fruit_options**.

Uma vez que criamos o corpo, precisamos adicionar este corpo no composto, agora o que é um composite (composto)?

Como você viu nas imagens, um composto consiste em vários corpos dentro de si. Quando queremos que vários corpos tenham as mesmas propriedades, como forma e tamanho, e se comportem de determinada maneira, fazemos um composto desses corpos.

Em nosso jogo, a corda que estamos criando é composta de vários retângulos, por isso a chamamos de **composite**. Mas também temos que adicionar nossas frutas na mesma composição.

Para adicionar um corpo ao composite, usamos a função **Matter.Composite.add(nome_do_composto, corpo_a_adicionar)**.

```
var fruit_options={
  density:0.001
}

fruit = Bodies.circle(300,300,15,fruit_options);
Matter.Composite.add(rope.body,fruit);
}
```

<p>Aqui, o composite é um corpo da classe Rope, e estamos adicionando a fruta a ele.</p>	
<p>Agora temos o corpo da fruta no composto, mas precisamos exibi-la também.</p> <p>Para exibir este corpo, criaremos um círculo usando a função ellipse na função draw().</p> <p>Para as posições x e y do círculo, passaremos os valores das posições x e y do corpo da fruta.</p> <p>Ao executar o código, a fruta aparecerá na posição definida e cairá no chão.</p> <p>Clique no botão Go live para ver o resultado.</p> <p>Porque não adicionamos nenhuma restrição entre o corpo da fruta e a corda. Eles fazem parte do mesmo composto, mas não estão vinculados um ao outro com nenhuma restrição.</p>	<p><i>O aluno executa o código e observa o resultado.</i></p>
<div data-bbox="183 1083 1105 1461"> <pre>function draw() { background(51); ground.show(); rope.show(); ellipse(fruit.position.x,fruit.position.y,15,15); Engine.update(engine); }</pre> </div> <p>Resultado:</p>	



Agora precisamos escrever o código para pendurar a fruta com a nossa corda.

Para isso, vamos usar restrições.

Se você se lembra, nossa corda é composta de vários retângulos, **6** retângulos para ser mais específico, que especificamos ao criar o objeto de corda.

Vamos criar uma restrição entre o corpo da fruta e o último retângulo de nossa corda.

Vamos programá-la!

Criamos a classe para a restrição porque nas classes futuras vamos pendurar a fruta com várias cordas.

Para deixar nosso código limpo, manteremos todo o código relacionado à criação da restrição no arquivo **link.js**.

Crie um novo arquivo chamado **link.js** e adicione-o ao arquivo **html** como fonte.

*O aluno cria um novo arquivo e o nomeia **link.js**.*

```
<!DOCTYPE html><html><head>
  <script src="p5.min.js"></script>
  <script src="p5.dom.min.js"></script>
  <script src="p5.sound.min.js"></script>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="utf-8">

</head>

<body>
  <script src="matter.min.js"></script>
  <script src="p5.play.js"></script>
  <script src="rope.js"></script>
  <script src="ground.js"></script>
  <script src="link.js"></script>
  <script src="sketch.js"></script>

</body></html>
```

A restrição será entre 2 corpos, o último corpo (retângulo) da corda e o corpo da fruta.

É por isso que, na classe Link, quando escrevemos o constructor, precisamos manter 2 parâmetros como **bodyA** e **bodyB**.

Queremos conectar o corpo da fruta ao último retângulo da corda. Então, vamos criar uma variável para obter o índice do último retângulo (**ou elemento, o composto pode ser considerado uma matriz**)

Isso será **var lastlink = bodyA.body.bodies.length-2**

O último elemento será **2** a menos que o comprimento porque o índice sempre começa em **0**, e também adicionamos frutas no composto que aumenta o comprimento **1**. Portanto, para obter o índice do último elemento, precisamos subtrair 2 do comprimento.

Então a função **bodyA** é **rope.body.bodies[lastlink]**.

bodyB é a fruta.

Criaremos a restrição usando a função **createConstraint()** da biblioteca **matter.js**.

Para conectar 2 corpos, também precisamos especificar em que ponto, no corpo da restrição, ela será conectada.

Para ambos os corpos, podemos dizer **pointA** e **pointB**, que são as **posições X e Y**, então, vamos escrevê-las como **x:0** e **y:0**.

Precisamos definir mais dois parâmetros:

- O comprimento da restrição, que também mantemos em 0 por enquanto.
- A rigidez, que escrevemos como **0.01**. Este valor de rigidez evitará a vibração da restrição (aqui, a vibração se refere aos pequenos movimentos na corda). Para que nossa restrição permaneça estável.

```
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);
  }
}
```

Assim que a restrição for concluída, adicionaremos ela ao nosso mundo.

Mas antes de executarmos o código, precisamos criar o objeto da classe **link()** no arquivo **sketch.js** e passar os 2 corpos.

Para criar o objeto, primeiro declare a variável como **var fruit_con;**

Na função **setup**, iremos criar o objeto e atribuí-lo à variável **fruit_con;**

Ao criar o objeto da classe, precisamos passar a corda e a fruta como parâmetros no objeto de classe **Link()**.

```
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);

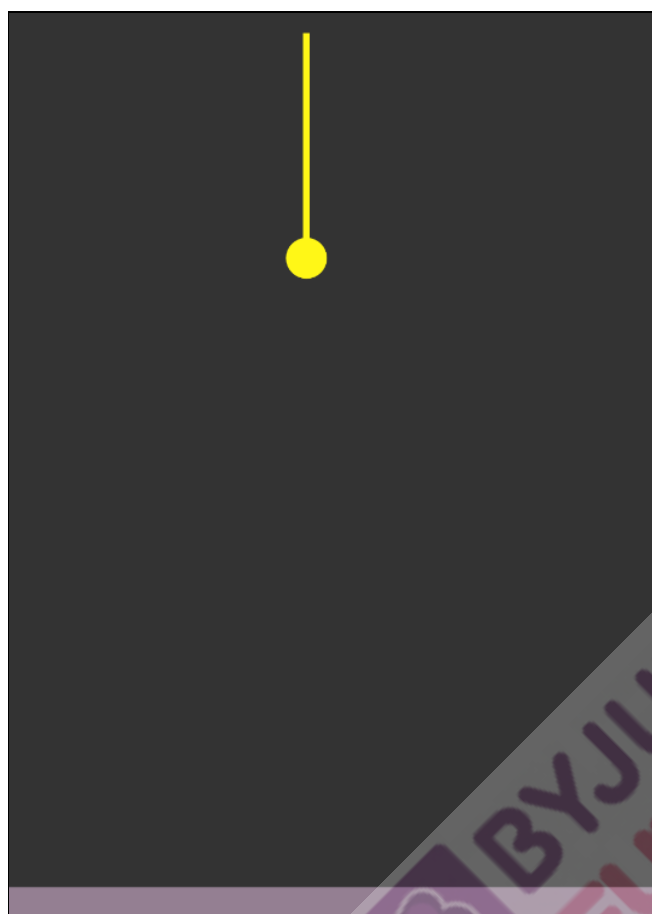
  fruit_con = new Link(rope,fruit);

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

Agora execute o código e veja o resultado.

Não é incrível?

O aluno executa o código e observa o resultado.



Terminamos por hoje.

Criamos a classe de base e o corpo da fruta e adicionamos uma corda à nossa tela enquanto prendemos o corpo da fruta a ela.

Na próxima aula, vamos adicionar a habilidade de partir a corda e fazer a fruta cair. Também criaremos nosso coelho.

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"> • Elogie o aluno. • Revise as atividades da aula atual. • Discuta os questionários. 	
Sessão de Perguntas e Respostas - Clique no questionário de aula	
Pergunta	Resposta
<p>Para tornar o solo um corpo estacionário, qual das seguintes propriedades é usada?</p> <p>A. isStatic:true B. isStatic:false C. isStatic = true D. isStatic = false</p>	A
<p>Qual função é usada para impedir que um corpo tenha um contorno?</p> <p>A. noStroke = true B. stroke(null) C. Stroke = null D. noStroke()</p>	D
<p>Qual das afirmações a seguir é verdadeira sobre um composto?</p> <p>A. Precisamos adicionar um corpo ao composto para aplicar suas propriedades. B. Um composto consiste em vários corpos contidos por ele. C. Quando queremos que vários corpos tenham as mesmas propriedades, como forma e tamanho, e se comportem de determinada maneira, fazemos um composto deles. D. Todas as alternativas acima.</p>	D

Encerrar o painel do questionário

FEEDBACK

- Incentive o aluno a fazer anotações para reflexão no formato Markdown.
- Elogie o aluno pelo seu esforço na aula.
- Revise o conteúdo da lição.

*** Este projeto levará apenas 30 minutos para ser concluído. Motive o aluno a tentar concluí-lo imediatamente após a aula.**

Panorama do Projeto

ESMAGUE OS ZUMBIS

Objetivo do Projeto:

Neste projeto, vamos criar as classes stone (pedra), base e link. Usando as restrições, junte a ponte com as articulações e empilhe pedras nela.

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 e 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 irá se sair muito bem.

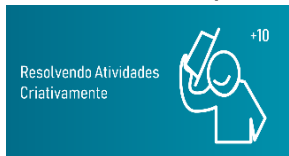
Até breve!




Os alunos se engajam no projeto com a professora.

Você recebeu uma Tirada de Chapéu

Vejo você na próxima aula.

Certifique-se de ter dado pelo menos 2 chapéus durante a aula para:



		 
<p>Professora encerra a apresentação de slides</p> 		
<p>Professora Clica em Terminar Aula</p>		
<p>ATIVIDADES ADICIONAIS</p>		
<p>Atividades Adicionais <i>Incentive o aluno a anotar suas reflexões no diário de reflexões usando o markdown.</i></p> <p>Use estas perguntas como orientadoras:</p> <ul style="list-style-type: none"> • O que aconteceu hoje? <ul style="list-style-type: none"> ◦ Descrever o que aconteceu. ◦ O código que escrevi. • Como me senti depois da aula? • O que eu aprendi sobre programação e desenvolvimento de jogos? • Quais aspectos da aula me ajudaram? Quais eu achei difíceis? 		<p><i>O aluno usa o editor markdown para escrever suas reflexões no diário de reflexões.</i></p>

Links:

Nome da Atividade	Descrição	Link
Atividade da Professora 1	Código modelo	https://github.com/ByjusBrazil/PRO_1-1_C29_AtividadeDaProfessora1
Atividade do Aluno 1	Código modelo	https://github.com/ByjusBrazil/PRO_1-1_C29_AtividadeDoAluno1
Atividade da Professora 2	Código de referência	https://github.com/ByjusBrazil/PRO_1-1_C29_AtividadeDaProfessora2
Solução do Projeto	Esmague os Zumbis	https://github.com/ByjusBrazil/PRO_1-1_C29_SolucaoDoProjeto
Link do recurso visual de referência da professora	Recurso visual	https://s3-whjr-curriculum-uploads.whjr.online/dccd523c-f975-4841-b2e0-180e1ae948ac.html
Referência da Professora do Questionário de aula	Questionário de aula	https://s3-whjr-curriculum-uploads.whjr.online/42a42d90-e6d7-4f8d-bafe-035cc58ea283.pdf
Apresentação Sob os Holofotes (Aluno)	Sob os Holofotes	https://s3-whjr-curriculum-uploads.whjr.online/0da11c9f-cf52-495d-bc20-b011eaf26adb.html
Apresentação Sob os Holofotes com anotações (Professora)	Sob os Holofotes	https://s3-whjr-curriculum-uploads.whjr.online/4ec6ec2f-b1ae-4afa-9734-bcd35d9b93ae.html
Link do Recurso Visual para Referência da Professora	Recurso visual	https://s3-whjr-curriculum-uploads.whjr.online/458cb017-b714-4ec3-a5ba-a5688d99e6d1.html