

Tópico	CONDICIONAIS - IF	
Descrição da aula	As crianças aprendem os princípios básicos da lógica, que são os condicionais na programação. Assim, explorando os diferentes ramos do fluxo lógico com base em condições, as crianças aplicam a lógica condicional para criar um jogo interativo.	
Aula	ADV-C12	
Duração	55 mins	
Objetivo 	<ul style="list-style-type: none"> Fazer o design e criar a lógica para o movimento do nosso jogador usando as setas do teclado 	
Recursos necessários	<ul style="list-style-type: none"> Recursos da Professora: <ul style="list-style-type: none"> Laptop/PC com Camera Login no Code.org Fone de ouvido com microfone Bloco de notas e caneta Recursos do Aluno: <ul style="list-style-type: none"> Laptop/PC com Camera Login no Code.org <ul style="list-style-type: none"> Aparece no painel Fone de ouvido com microfone (Opcional) Bloco de notas e caneta 	
Estrutura da aula	Introdução Atividade Dirigida pela Professora Atividade Dirigida pelo Aluno Fechamento Dicas e Sugestões do Projeto	5 Min 10 Min 30 Min 5 Min 5 Min
SESSÃO DE INTRODUÇÃO - 2 min		



A professora inicia apresentação de slides dos slides 1 a 48.
Consulte as anotações da oradora e siga as instruções em cada slide.



ATIVIDADE DA PROFESSORA - 8 min

A Professora Inicia o Compartilhamento de Tela

Dizer



Fazer



Demonstração do jogo completo

O jogador (Astronauta verde) está se movendo para cima, para baixo, para a esquerda e para a direita usando as setas do teclado OU no celular ele será movido pelo joystick.

Fique longe dos inimigos (astronauta vermelho e preto). Se o jogador tocar em qualquer um dos inimigos, a pontuação de vida será reduzida em 0,5. O jogador tem apenas 10 vidas.

Lembre-se, quando você tocar em qualquer parte da borda da tela do jogo, os sprites irão se recuperar.

Abra a [Atividade da Professora 1- VÍDEO COMPLETO DO RESULTADO.](#)

Abra o vídeo e mostre o jogo que concluiremos na próxima aula C13.

Abra a [Atividade da Professora 2- DIAGRAMA DO CÓDIGO](#)

Use este documento enquanto explica a atividade da aula, ele ajudará o aluno a visualizar

Demonstração da saída da aula atual -

O sprite do Jogador está se movendo para cima, para baixo, para a esquerda e para a direita usando as teclas direcionais do teclado OU no celular ele será movido pelo joystick.

Como não escrevemos código para o que acontece a seguir, se o jogador ou os inimigos baterem ou tocarem na borda, irão passar e ir além da tela de jogo.

Portanto, agora o jogo parece estar incompleto. A razão é que iremos completar este jogo na próxima aula.

Então, abra a [Atividade da Professora 3 - RESULTADO DA AULA ATUAL](#) para mostrar ao aluno qual parte do jogo será construída na aula de hoje.

Antes de começarmos, deixe-me explicar rapidamente para você sobre o bloco **IF** e por que ele é usado.

Por exemplo: seus pais já disseram que se você tirar boas notas em uma prova, eles vão lhe dar uma bicicleta ou algum outro presente?

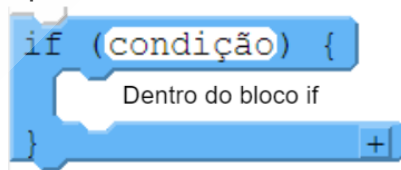
OU

Sua mãe já disse que, se você a ajudar na limpeza da casa, ela vai lhe dar seu lanche favorito?

Esses são exemplos de uma condição **IF**. Ou seja, se você satisfizer a condição de ajudar sua mãe ou de obter boas notas, só assim ganhará seu lanche ou presente predileto.

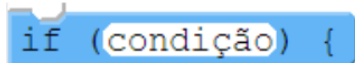
Da mesma forma, na programação, temos uma condição IF que verifica se uma condição está sendo satisfeita. Se estiver satisfeita, deve executar apenas a tarefa mencionada.

Então é assim que o bloco if se parece:



Você consegue esse bloco na seção **Control**.

Este bloco de código verifica a condição que está escrita entre parênteses ()



O código dentro do bloco if é executado apenas se a condição for verdadeira.

Exemplo:

```
if ( 10 == 10 ) {  
  //Qualquer código que é mencionado dentro da condição if  
}
```

Agora, neste exemplo, a condição **if** será sempre verdadeira porque o número 10 é sempre igual ao número 10. Portanto, essa condição sempre será verdadeira e executará qualquer código que esteja escrito dentro dela.

Passo 2: Faça o design do jogo

Mantendo o propósito do jogo em mente, o design do jogo exigirá os seguintes recursos:

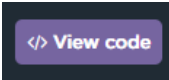
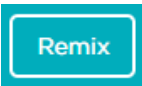
- Precisamos criar 4 animações:
 1. Um plano de fundo
 2. Um Astronauta Verde (jogador)
 3. Um Astronauta Vermelho (inimigo 1)
 4. Um Astronauta Preto (inimigo 2)



OBSERVAÇÃO: a professora deve completar o código que está em **Atividade da Professora 4** enquanto explica as instruções abaixo.

Na **Atividade da Professora 4**, alguns dos códigos e animações são fornecidos previamente a você.

View codeRemix

Abra a **Atividade da Professora 4** e clique em  e depois em  para que você possa editar o código. Comece explicando o código de acordo com as instruções fornecidas abaixo.

Os passos para criar uma nova animação são fornecidos abaixo.

Como todas as animações já foram criadas, exceto a animação para o astronauta preto (inimigo 2), irei começar a criar a animação para o astronauta preto. Antes disso, irei baixar o astronauta preto para o meu sistema para poder carregá-lo no Laboratório de Jogos. Todos os recursos são fornecidos a mim em **Atividade da Professora 5** e, para você, é fornecido em **Atividade do Aluno 3**.

1. Para criar uma animação, primeiro clique em **Animação..**



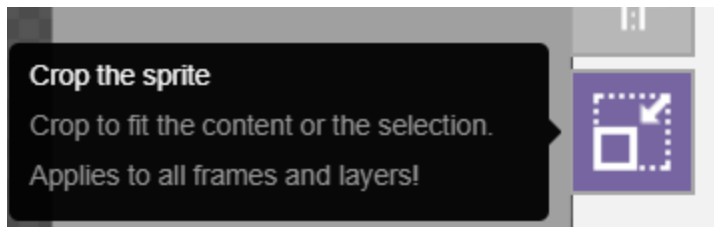
2. Então, clique em **nova animação**.



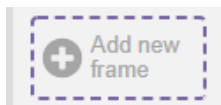
3. Agora, clique no botão **Enviar imagem** e envie a imagem do astronauta preto do seu computador.



4. Depois de carregado, clique no ícone **Cortar o sprite (Crop the sprite)** para remover o espaço em branco extra ao redor do astronauta.



5. Como queremos mostrar uma animação para um astronauta ambulante, precisamos adicionar mais uma imagem, de modo que a reprodução dessas duas imagens consecutivas crie uma animação. Para enviar outra imagem, clique na guia **Adicionar nova moldura (Add new frame)**.



6. Envie a segunda imagem do astronauta preto e recorte-a para remover o espaço em branco extra.



Enviar imagem

7. Agora, renomeie a animação para **inimigo_2**.



A criação da animação está terminada.

Passo 3: Código

O código completo para hoje:

```
var plano_de_fundo = createSprite(200, 200);  
plano_de_fundo.setAnimation(▼"plano_de_fundo");  
  
var jogador = createSprite(200, 200);  
jogador.setAnimation(▼"jogador");  
jogador.scale = 0.13;  
  
var inimigo_1 = createSprite(20, 20);  
inimigo_1.setAnimation(▼"inimigo_1");  
inimigo_1.scale = 0.13;  
inimigo_1.velocityX = 1;  
inimigo_1.velocityY = 2;  
  
var inimigo_2 = createSprite(350, 20);  
inimigo_2.setAnimation(▼"inimigo_2");  
inimigo_2.scale = 0.13;  
inimigo_2.velocityX = 3;  
inimigo_2.velocityY = 4;  
  
createEdgeSprites();
```

```
function draw() {
  drawSprites();
  if (keyDown(▼ "left")) {
    jogador.velocityX = -5;
  }
  if (keyDown(▼ "right")) {
    jogador.velocityX = 5;
  }
  if (keyDown(▼ "up")) {
    jogador.velocityY = -5;
  }
  if (keyDown(▼ "down")) {
    jogador.velocityY = 5;
  }
}
```

Explicação:

- 1. Criando o sprite:** primeiro, vamos criar todos os sprites necessários. Você pode ver que eu já criei sprites para o plano de fundo, para o jogador e para o inimigo_1. Só preciso criar um sprite para o inimigo_2 e definir suas propriedades.

```
var inimigo_2 = createSprite(350, 20);
inimigo_2.setAnimation(▼ "inimigo_2");
inimigo_2.scale = 0.13;
inimigo_2.velocityX = 3;
inimigo_2.velocityY = 4;
```

a. Crie o sprite inimigo_2:

Para isso, primeiro, vou arrastar e soltar este bloco de código


```
var sprite = createSprite(200, 200);
```

da seção **Sprites**, e renomear o nome da variável de **sprite** para **inimigo_2**.

```
var inimigo_2 = createSprite(350, 20);
```

Agora, defina $X = 350$ e $Y = 20$. Você pode definir o sprite para qualquer posição de sua escolha. Apenas certifique-se de que está longe do sprite do jogador. Porque se o sprite inimigo tocar o sprite do jogador no próprio início do jogo, a vida do jogador começará a ser reduzida desde o início do jogo.

b. Defina a imagem do inimigo_2:

Para isso, arraste e solte este bloco de código

```
sprite.setAnimation(label)
```

da seção

Sprites

, e renomeie o nome da variável de **sprite** para **inimigo_2**. Além disso, defina a animação para o **inimigo_2**.

```
inimigo_2.setAnimation("inimigo_2");
```

c. Defina o tamanho do inimigo_2:

Para definir o tamanho do sprite, use o bloco **sprite.scale**. Escala significa definir o tamanho da imagem. O valor da escala pode ser qualquer coisa entre 0 e 1. Precisamos renomear o nome da variável de **sprite** para **inimigo_2** e definir a escala para 0,13, pois não queremos que o sprite seja muito grande.

```
inimigo_2.scale = 0.13;
```

d. Defina a velocidade do inimigo_2 na direção X:

Para isso, arraste e solte este bloco de código

`sprite.velocityX` da seção **Sprites** e

renomeie o nome da variável de **sprite** para **inimigo_2**. E definimos a velocidade na direção X. Essa é a velocidade na qual o sprite se moverá na direção X. Vamos definir a velocidade para 3.

```
inimigo_2.velocityX = 3;
```

- e. Da forma como definimos a velocidade de X, precisamos definir a velocidade de Y.

```
inimigo_2.velocityY = 4;
```

Esta é a velocidade na qual o sprite se moverá na direção Y. Vamos definir a velocidade para 4.

2. Escreva a função: Agora vamos escrever o código dentro da função

`function draw() {` . Esta função será executada continuamente.

- Use o bloco `drawSprites();` para colocar/exibir todos os sprites na tela do jogo.
- Verifique se a tecla de seta para a **esquerda** do teclado foi pressionada usando a condição if. Portanto, a condição deve ser: se a tecla de seta para a **esquerda** do teclado for pressionada, precisamos mover nosso jogador (astronauta verde) na direção esquerda. Portanto, adicione uma condição **if** dentro da função **draw**.

```
function draw() {
  drawSprites();
  if ( ) {
    +
  }
}
```

Então, dentro dos parênteses (), coloque este bloco

`keyDown (code)`

Você obterá este formulário na seção do mundo

World

Este bloco `keyDown (code)` verifica se a tecla especificada foi pressionada ou não.

Como a chave que estávamos procurando é para a esquerda, o código dentro da função **keyDown()** deve ser definido como **left**.

`keyDown (▼ "left")`

Portanto, a condicional **if** deve ser semelhante a esta:

```
function draw() {
  drawSprites();
  if ( keyDown (▼ "left") )
```

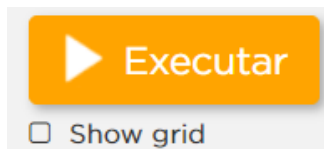
E dentro desse bloco **if**, precisamos definir o valor da velocidade para a direção X em **-5**.

Agora, vamos entender por que estamos atribuindo **-5**.

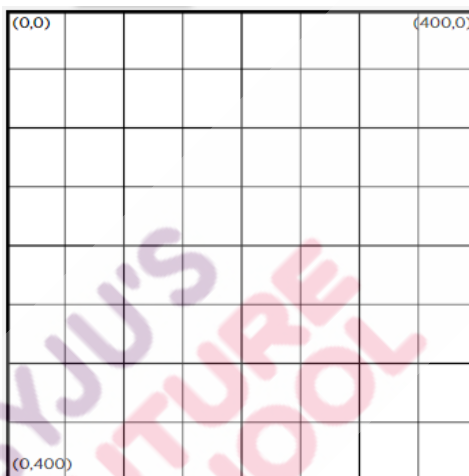
A seguir está a grade do Laboratório de jogos.

OBSERVAÇÃO: para ligar e desligar essa grade, use a

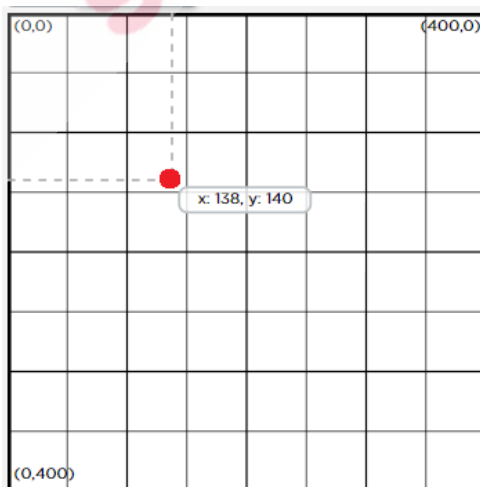
caixa de seleção mencionada abaixo do botão 'Executar'.



Assim que a caixa de seleção "Show Grid" (Mostrar grade) estiver ativada, a tela terá a seguinte aparência:

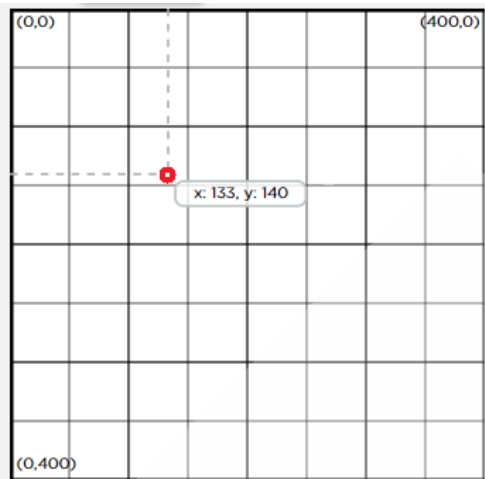


Suponha que estejamos na posição $x = 138$ e $y = 140$.



E queremos ir para a esquerda. Portanto, precisamos reduzir o valor do eixo X. Agora, depende de até onde queremos ir.

Se quisermos mover 5 pixels, precisamos reduzir 5 unidades do valor do eixo X. Portanto, como resultado, deve ser $138 - 5 = 133$.



Então, subtraindo 5 da velocidade de X, o sprite pode ser movido para a esquerda.

Então, primeiro, arraste e solte este bloco

`sprite.velocityX`

e o renomeie para **jogador**, porque queremos mover nosso jogador e definir o valor para **-5**.

`jogador.velocityX = -5;`

Aqui, o **jogador** é o astronauta verde que se moverá para a esquerda se pressionarmos a tecla de seta para a esquerda no teclado.

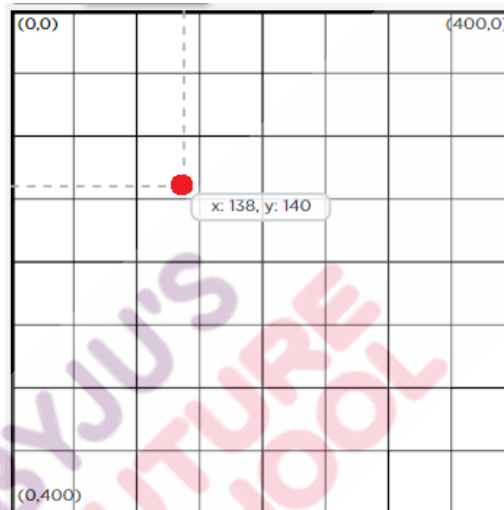
```
function draw() {
  drawSprites();
  if (keyDown("left")) {
    jogador.velocityX = -5;
  }
}
```

c. Da mesma forma, verifique se a tecla de seta para a direita está

pressionada.

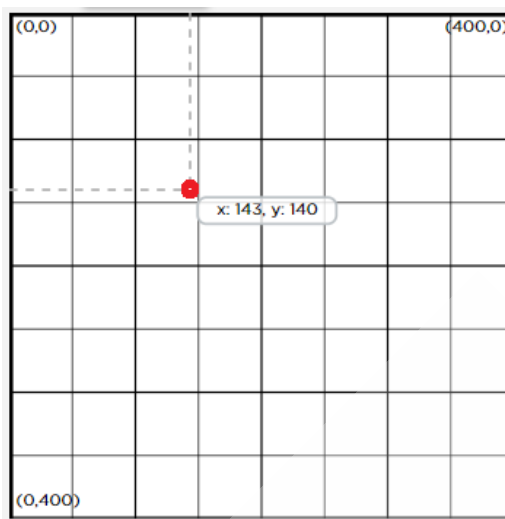
```
if ( keyDown (▼ "right") ) {  
    
}
```

Suponha que estejamos na posição $x = 138$ e $y = 140$.



E queremos ir para a direita. Portanto, temos que adicionar valores ao eixo X. Agora, depende de até onde queremos ir. Se quisermos mover 5 pixels para o lado direito, precisamos adicionar 5 ao valor do eixo X. Portanto, será $138 + 5 = 143$.

Portanto, para mover 5 pixels para a direita, temos que adicionar 5 à Velocidade de X.

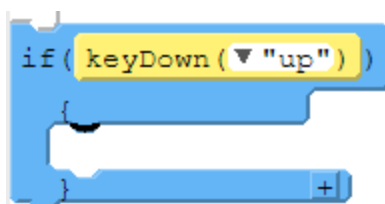


Portanto, ao adicionar 5 à velocidade de X, o sprite se moverá para o lado direito.

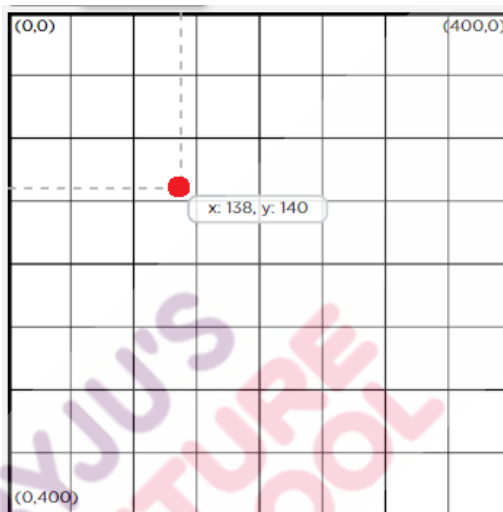
```
jogador.velocityX = 5;

function draw() {
  drawSprites();
  if (keyDown(↴ "left"))
  {
    jogador.velocityX = -5;
  }
  if (keyDown(↴ "right"))
  {
    jogador.velocityX = 5;
  }
}
```

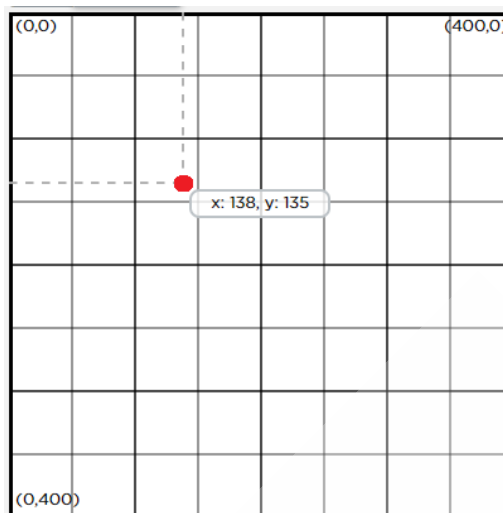
- d. Da mesma forma, verifique se a tecla de seta para cima está pressionada.



Suponha que estejamos na posição $x = 138$ e $y = 140$.



E queremos nos mover para cima. Portanto, temos que reduzir o valor do eixo Y. Agora, isso depende de até onde queremos ir. Se quisermos mover 5 pixels, precisamos reduzir 5 do valor do eixo Y. Portanto, será $140 - 5 = 135$.
Portanto, para mover 5 pixels para cima, temos que subtrair 5 da Velocidade Y do jogador.

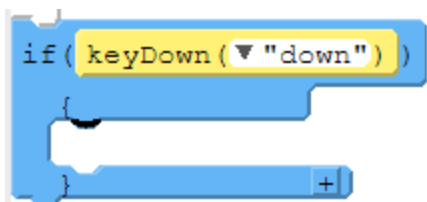


Então, subtraindo 5 da velocidade de Y, o sprite irá se mover para cima.

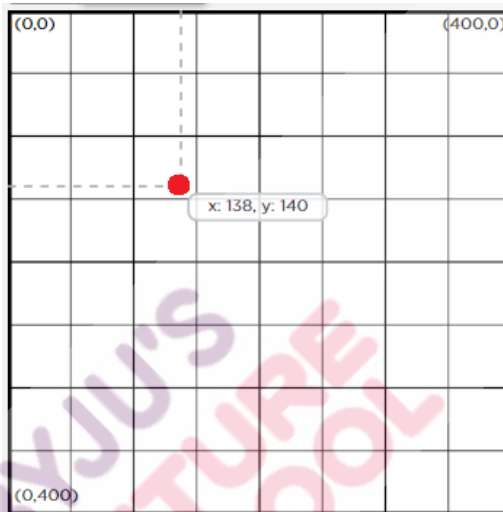
```
jogador.velocityY = -5;

function draw() {
  drawSprites();
  if (keyDown(↵"left"))
  {
    jogador.velocityX = -5;
  }
  if (keyDown(↵"right"))
  {
    jogador.velocityX = 5;
  }
  if (keyDown(↵"up"))
  {
    jogador.velocityY = -5;
  }
}
```

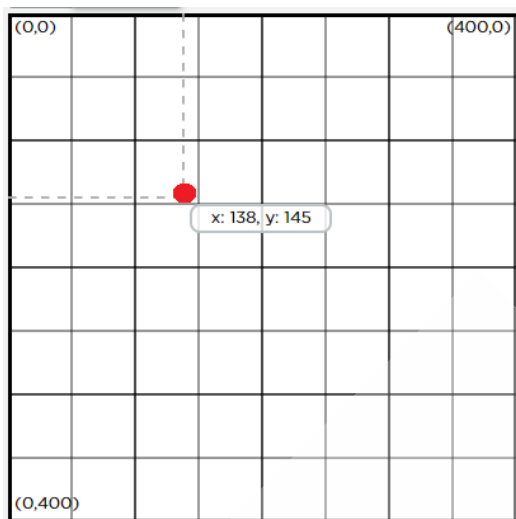
- e. Da mesma forma, verifique se a tecla de seta para baixo está pressionada



Suponha que estejamos na posição $x = 138$ e $y = 140$.



E queremos nos mover para baixo. Portanto, temos que adicionar valores ao eixo Y. Agora, isso depende de até onde queremos ir. Se quisermos mover 5 pixels, precisamos adicionar 5 ao eixo Y. Portanto, será $140 + 5 = 145$.
Portanto, para mover 5 pixels para baixo, temos que adicionar 5 à velocidade de Y.



Então, adicionando 5 à velocidade de Y, o sprite irá se mover para baixo.

```
jogador.velocityY = 5;
```

```
function draw() {  
  drawSprites();  
  if (keyDown(▼ "left"))  
  {  
    jogador.velocityX = -5;  
  }  
  if (keyDown(▼ "right"))  
  {  
    jogador.velocityX = 5;  
  }  
  if (keyDown(▼ "up"))  
  {  
    jogador.velocityY = -5;  
  }  
  if (keyDown(▼ "down"))  
  {  
    jogador.velocityY = 5;  
  }  
}
```

Passo 4: Testar o jogo

Portanto, o 4º passo é testar o jogo. Deixe-me apertar o botão para testar o jogo. Assim que pressionar as teclas de seta, o sprite começa a se mover de acordo com elas.

 Executar

A Professora Para de Compartilhar a Tela

ATIVIDADE DO ALUNO - 30 min

O Aluno Inicia o Compartilhamento de Tela

- 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 modo de Tela Cheia

Dizer



Agora é sua vez.
Você tem que construir o mesmo jogo.

Você pode me dizer os passos que vai seguir?

Passo 1: Definir uma finalidade

Neste passo, especificamos o que queremos que o jogo faça.

Passo 2: Fazer o design do jogo

Neste passo, especificamos os componentes de design que queremos em nosso jogo.

Passo 3: Programar o jogo

Neste passo, especificamos os blocos de código que queremos usar em nosso código.

Passo 4: Testar o jogo

Neste passo, rodamos o jogo para verificar se funciona.

Peça ao aluno para fazer o seguinte:


Fazer



[Atividade do Aluno 1-
PROGRAMAR O JOGO](#)

Peça ao aluno para entrar no code.org e abrir a [Atividade do Aluno -1](#).

Clique em

 Ver código

e então clique em

 Remix

e comece a construir seu jogo.

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

[Atividade do Aluno 3-
RECURSOS](#)

Peça ao aluno para baixar os recursos da [Atividade do Aluno - 3](#)


1. Crie animações para ambos os inimigos.
2. Crie os sprites inimigos e o jogador, e defina as animações para todos os 3 sprites.
3. Defina a velocidade X e Y dos inimigos e defina o tamanho de ambos os sprites.
4. Dentro da função draw, siga as seguintes etapas:
 - a. Adicione a função drawSprite().
 - b. Programe o jogo para verificar se a seta para a esquerda está pressionada. Então, dentro do bloco IF, escreva o código para mover o sprite do jogador para a esquerda.
 - c. Programe o jogo para verificar se a seta para a direita foi pressionada. Então, dentro do bloco IF, escreva o código para mover o sprite do jogador para a direita.
 - d. Programe o jogo para verificar se a seta para baixo foi pressionada. Então, dentro do bloco IF, escreva o código para mover o sprite do jogador para baixo.
 - e. Programe o jogo para verificar se a seta para cima foi pressionada. Então, dentro do bloco IF, escreva o código para mover o sprite do jogador para cima.

Peça novamente ao aluno para explicar os passos. Ajude o aluno se necessário.


Peça ao aluno para copiar o link do projeto e colá-lo no campo **Link do projeto no painel do aluno** e clicar em **Enviar**.

A Professora Orienta Aluno a Parar de Compartilhar a Tela

SESSÃO DE FECHAMENTO - 5 min

A professora inicia apresentação de slides  **do slide 49.**
Consulte as anotações da oradora e siga as instruções em cada slide.

DICAS E SUGESTÕES PARA O PROJETO - 5 min

A professora inicia apresentação de slides  **do slide 50.**
Consulte as anotações da oradora e siga as instruções em cada slide.

A Professora Inicia o Compartilhamento de Tela

Dizer



Fazer



LANÇAMENTO DE FOGUETE

Objetivo do projeto:

Hoje, aprendemos sobre a condicional IF e os eventos do teclado e os usamos para construir um jogo.




Neste projeto, você tem um sprite de foguete que deve ser lançado quando a barra de espaço for pressionada, se mover em direção ao planeta e parar quando atingir o planeta.

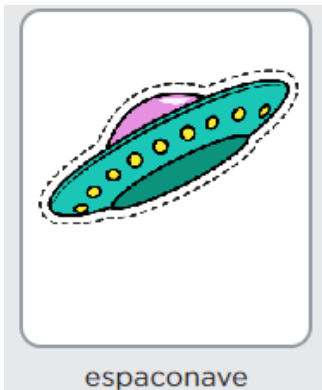
História:

Os alunos da escola St. Fairy estão aprendendo sobre diferentes planetas e foguetes. Então, o professor da escola quer que você crie um pequeno jogo interativo no qual o foguete é lançado da Terra, viaja no espaço e pousa em outro planeta.

Você consegue criar um jogo assim?

Abra o link da **Resolução do Projeto** e faça uma demonstração do projeto para o aluno

Estou muito animada para ver a solução do seu projeto e sei que você se sairá muito bem.	
Até breve!	
<p>A professora inicia apresentação de slides  do slide 51. Consulte as anotações da oradora e siga as instruções em cada slide.</p>	
Professora Clica em	× Terminar Aula
Atividades Adicionais	
<p>A professora inicia apresentação de slides  do slide 51 a 55. Consulte as anotações da oradora e siga as instruções em cada slide.</p>	
ATIVIDADE ADICIONAL DO ALUNO 	
O Aluno Inicia o Compartilhamento de Tela	
<p>Atividade Adicional 1:</p> <p>Adicione um recurso a esta animação. Quando a tecla A do teclado for pressionada, uma nave espacial deve aparecer.</p> <p>Escreva o seguinte código dentro da função draw:</p> <ol style="list-style-type: none"> 1. Crie uma animação para a nave espacial. Esta animação deve ser exibida quando o jogador ou o usuário pressiona a tecla A. <p>Observação: peça ao aluno para encontrar uma imagem de nave espacial na Biblioteca de Animação do code.org e selecionar aquela que ele gostar.</p>	



2. Verifique se a tecla “a” do teclado foi pressionada. Escreva o seguinte código dentro do bloco IF.

```
if (keyDown(▼ "a"))
{
```

- a. Crie um sprite na posição x e y aleatória e defina sua animação para a animação da nave espacial que você acabou de criar.

```
var espaconave = createSprite(randomNumber(10, 300), randomNumber(10, 300));
espaconave.setAnimation(▼ "espaconave");
```

- b. Defina o tamanho do sprite.

```
espaconave.scale = 0.20;
```

- c. Em seguida, defina a velocidade do sprite ao longo do eixo X, que é horizontal.

```
espaconave.velocityX = 1;
```

- d. Em seguida, defina a velocidade do sprite ao longo do eixo Y, que é vertical.

```
espaconave.velocityY = 2;
```

Código completo da AA:

```

if (keyDown(▼"a")) {
  var espaconave = createSprite(randomNumber(10, 300), randomNumber(10, 300));
  espaconave.setAnimation(▼"espaconave");
  espaconave.scale = 0.20;
  espaconave.velocityX = 1;
  espaconave.velocityY = 2;
}
  
```

Número da atividade	Nome da atividade	Links
Atividade da Professora 1	VÍDEO COMPLETO DO RESULTADO	https://s3-whjr-curriculum-uploads.whjr.online/2c02cb87-5393-4490-80cc-ebb3165853f7.mp4
Atividade da Professora 2	DIAGRAMA DO CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/bab7cf7f-206b-4738-8dc4-4a8697893a61.pdf
Atividade da Professora 3	VÍDEO DO RESULTADO DA AULA ATUAL	https://s3-whjr-curriculum-uploads.whjr.online/226937e3-8172-4ce9-befa-52b576d849af.mp4
Atividade da Professora 4	PROGRAME O JOGO	https://studio.code.org/projects/gamelab/UecIDqWMnLy0fb6yoE2jNcPxSTrYwXyYOv9Cpikk4
Atividade da Professora 5	RECURSOS	https://drive.google.com/drive/folders/1wDRO-REG5MmptFi3A91e4J9JVYcfq02S?usp=sharing
Atividade do Aluno 1	PROGRAME O JOGO	https://studio.code.org/projects/gamelab/MlnptnlfQLoVPvXlu9aszW8TNmW7w0bAjV9paOQd7c

Atividade do Aluno 2	DIAGRAMA DO CÓDIGO	https://s3-whjr-curriculum-uploads.whjr.online/bab7cf7f-206b-4738-8dc4-4a8697893a61.pdf
Atividade do Aluno 3	RECURSOS	https://drive.google.com/drive/folders/1wDR O-REG5MmptFi3A91e4J9JVYcfq02S?usp=sharing
Resolução do projeto	LANÇAMENTO DE FOGUETE	https://studio.code.org/projects/gamelab/HH TIHijn1dLCPs1IEW-0sWHuLlupqXghDcvkrdl Km5Y Aperte a barra de espaço para lançar o foguete
Link do recurso visual para referência da professora	Link do recurso visual	https://s3-whjr-curriculum-uploads.whjr.online/ed9005d9-2e44-491f-a509-487e809eb10d.html
Referência da professora: Questionário de aula	Questionário de aula	https://s3-whjr-curriculum-uploads.whjr.online/ffe07fe1-b067-4e20-b884-fec2c86db28c.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/6b2ea20c-7759-407c-b27c-dbe1a93e319d.html