

Tópico	CONDICIONAIS	
Descrição da aula	As crianças aprendem a aplicar condições na lógica do programa para criar um jogo sem fim, interativo e emocionante, reforçando sua confiança e suas habilidades de resolução de problemas.	
Aula	ADV-C16	
Tempo da aula	55 min	
Objetivo 	<ul style="list-style-type: none"> • Completar o jogo de carro sem fim. 	
Recursos Necessários	<ul style="list-style-type: none"> • Recursos da professora: <ul style="list-style-type: none"> ○ Laptop/PC com WebCam ○ Credenciais para login no Code.org ○ Fones de ouvido com microfone ○ Caderno de anotações e caneta • Recursos do aluno: <ul style="list-style-type: none"> ○ Laptop/PC com WebCam ○ Credenciais para login no Code.org ○ Fones de ouvido com microfone (opcional) ○ Caderno de anotações e caneta 	
Estrutura da aula	Introdução Atividade Dirigida pela Professora Atividade Dirigida pelo Aluno Fechamento Dicas e Sugestões para o Projeto	5 min 10 min 30 min 5 min 5 min
SESSÃO DE INTRODUÇÃO - 5 min		
<p>A professora inicia a apresentação de slides  do slide 1 ao 19. Consulte as anotações da oradora e siga as instruções de cada slide.</p>		

ATIVIDADE DA PROFESSORA



- 10 min

A Professora Inicia o Compartilhamento de Tela

Dizer



Nesta aula, completaremos o jogo que começamos a construir na última aula.

Faça a demonstração do jogo completo.



Pressione a **barra de espaço** para iniciar o jogo.

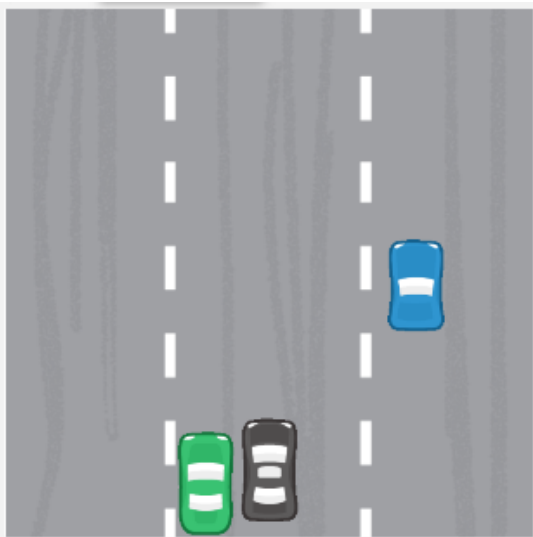
Fazer



Faça a demonstração do jogo completo.

Execute a

[Atividade-da-Professora-1.](#)



Uma vez que o jogo é iniciado, pressione as teclas das setas **esquerda** e **direita** para controlar seu carro e evite que ele seja atingido pelos outros carros.

Caso seja atingido por outro carro, o jogo será interrompido e você deve ver uma tela com a mensagem – “CLIQUE PARA RECOMEÇAR”.



Se clicar em qualquer lugar da tela, o jogo irá recomeçar.

Ótimo!

Agora que entendeu o objetivo, vamos começar a desenvolver esse jogo.

Passo 2: Criar o Design do Jogo

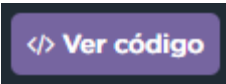
Tendo o objetivo do jogo em mente, o design do jogo deve exigir os seguintes recursos:

1. Uma imagem inicial (que aparece quando o jogo é iniciado)
 2. Uma imagem de fundo (a estrada)
 3. Um carro adversário verde
 4. Um carro adversário azul
 5. Um carro do jogador
 6. Uma imagem de 'Fim de Jogo' - o fundo de reinicialização (que aparece se perdermos o jogo)
- Então, precisamos criar mais três animações:
 1. Uma animação para a inicialização
 2. Uma animação para a reinicialização
 3. Uma animação para o carro do jogador



OBSERVAÇÃO: A professora deve completar o código presente na **Atividade-da-professora-2** enquanto explica as instruções abaixo.

Na **Atividade-da-professora-2**, alguns códigos e animações são dados a você previamente.



Abra **Atividade-da-professora-2** e clique em



para que o código possa ser editado. Comece explicando o código a partir das instruções dadas abaixo.

Os passos para criar uma nova Animação são dados abaixo:

Como já criei todas as animações exceto a do fundo de inicialização, agora vou começar a criar a sua animação. Antes disso, farei o download da imagem do fundo de inicialização (Game Bg-03-2.png) no meu sistema para que eu possa carregá-la no Laboratório de Jogos. Todos os recursos estão disponíveis na [Atividade-da-professora-3](#) para mim; para você, estão disponíveis na [Atividade-do-aluno-2](#).

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



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



3. Agora, clique no botão **Carregar imagem** e carregue a imagem do fundo de inicialização (Game Bg-03-2.png) do seu computador.



4. Agora, mude o nome da animação para **começar**.



Terminamos de criar a animação.

Passo 3: Programar as Regras

Observação: os códigos marcados nas imagens abaixo em VERMELHO são os novos códigos a serem adicionados pelo aluno nessa aula.

O código completo de hoje:

```
1  var estrada = createSprite(200, 200) → ;  
2  estrada.setAnimation(▼"estrada");  
3  estrada.scale = 1;  
4  
5  var carro_do_jogador = createSprite(200, 350) → ;  
6  carro_do_jogador.setAnimation(▼"carro");  
7  carro_do_jogador.scale = 0.6;  
8  
9  var carrol = createSprite(100, 200) → ;  
10 carrol.setAnimation(▼"carrol");  
11 carrol.scale = 0.6;  
12
```

```
13 var carro2 = createSprite(325, 350) → ;  
14 carro2.setAnimation(▼ "carro2") ;  
15 carro2.scale = 0.6;  
16  
17 var começar = createSprite(200, 200) → ;  
18 começar.setAnimation(▼ "começar") ;  
19  
20 var recomçar = createSprite(200, 200) → ;  
21 recomçar.setAnimation(▼ "recomeçar") ;  
22 recomçar.scale = 0.8;  
23 recomçar.visible = falso ;  
24  
25 createEdgeSprites();  
26
```



```
27 function draw() {  
28   drawSprites();  
29   if (keyDown(▼ "right")) {  
30     carro_do_jogador.velocityX = 3;  
31   }  
32   if (keyDown(▼ "left")) {  
33     carro_do_jogador.velocityX = -3;  
34   }  
35  
36   if (carro_do_jogador.isTouching(carro1)) {  
37     recomeçar.visible = verdadeiro;  
38   }  
39   if (carro_do_jogador.isTouching(carro2))  
40   {  
41     recomeçar.visible = verdadeiro;  
42   }
```

```

43  if( carro1.isTouching(carro2) ) {
44      carro1.x = randomNumber(400, 0);
45      carro1.y = 0;
46  }
47  if( carro1.y > 405 ) {
48      carro1.x = randomNumber(400, 0);
49      carro1.y = 0;
50  }
51  if( carro2.y > 405 ) {
52      carro2.x = randomNumber(400, 0);
53      carro2.y = 0;
54  }
55  carro_do_jogador.bounceOff( edges );

56  if( keyDown(▼"space") ) {
57      carro1.velocityY = 10;
58      carro2.velocityY = 10;
59      recomeçar.visible = falso;
60  }
61  if( mousePressedOver( recomeçar ) ) {
62      recomeçar.visible = falso;
63  }
64

```

Explicação:

1. **Criar o sprite:** Primeiro, vamos criar todos os sprites necessários. Você pode ver que eu já criei os sprites para criar o carro do jogador e o fundo de reinicialização. Só preciso criar um sprite para o fundo de inicialização e definir suas propriedades.

Observação: ao criar um sprite, a sequência na qual ele foi criado é sempre muito importante. Por exemplo: se você criar o sprite `carro_do_jogador` primeiro e só depois adicionar o código para criar o sprite de reinicialização, o sprite de reinicialização será colocado sobre o sprite `carro_do_jogador` e o sprite do carro não ficará visível, já que o sprite de reinicialização cobrirá completamente a tela.

Uma vez que o sprite de inicialização deve ficar visível antes do jogo começar, tudo bem se você criar o sprite de inicialização primeiro. Mas lembre-se de criar o sprite de reinicialização ao final, já que esse sprite deve ficar visível apenas se o jogador perder o jogo.



Sequência na qual os sprites devem ser criados:

1. Sprite da estrada.
2. Sprites dos carros (carros azul e verde) - qualquer um desses sprites pode ser colocado a seguir, não é necessário seguir uma ordem nesse caso. Você pode posicionar tanto o carro azul quanto o carro verde primeiro.
3. Sprite do carro do jogador.
4. Sprite do plano de fundo.
5. Sprite do fundo de reinicialização.

```
13 var carro2 = createSprite(325, 350) → ;
14 carro2.setAnimation(▼ "carro2" );
15 carro2.scale = 0.6;
16
17 var recomeçar = createSprite(200, 200) → ;
18 recomeçar.setAnimation(▼ "recomeçar" );
19 recomeçar.scale = 0.8;
20 recomeçar.visible = falso ;
21
22 var começar = createSprite(200, 200) → ;
23 começar.setAnimation(▼ "começar" );
```

a. **Criar o sprite de inicialização:**

Para isso, primeiramente vou arrastar e soltar esse bloco de

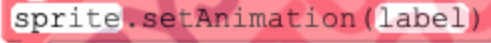

código  da
seção de  e mudar o nome da variável do
sprite para **começar**.

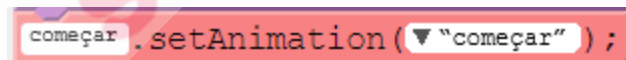


Este sprite será mostrado quando o jogo começar.

b. **Definir a imagem do fundo de inicialização:**

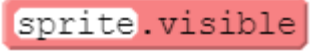
Para isso, arraste e solte esse bloco de código

 da seção 
e mude o nome da variável do **sprite** para **começar**. Além disso,
defina a animação para **começar** já que **começar** já possui um
fundo de inicialização.



2. **Visibilidade do fundo de reinicialização:**

Antes de continuar, vou falar sobre o bloco de código visível. Eu tenho o código para criar um sprite de reinicialização. Inicialmente, vamos definir esse sprite como **falso** já que não queremos mostrar esse sprite quando o usuário estiver jogando o jogo. Esse plano de fundo deve ficar visível apenas se o jogador bater em outro carro.

Para definir a visibilidade do sprite, use o bloco de código .
Esse bloco de código pode ser definido como **verdadeiro** ou **falso**. Se definirmos
esse bloco de código como **verdadeiro**, então o sprite se torna visível; se definirmos
como **falso**, o sprite não ficará visível.

```
recomeçar.visible = falso ;
```

3. **Evitar que os sprites saiam da tela do jogo:** precisamos acionar a função `createEdgeSprites()` para controlar as bordas da tela do jogo.

```
13 var carro2 = createSprite(325, 350) → ;
14 carro2.setAnimation(▼ "carro2" );
15 carro2.scale = 0.6;
16
17 var começar = createSprite(200, 200) → ;
18 começar.setAnimation(▼ "começar" );
19
20 createEdgeSprites();
```

Por fim, isso irá nos ajudar a evitar que os sprites saiam da tela do jogo. Essa função cria uma variável chamada 'bordas', a qual nos ajuda a manter o controle das bordas. Usaremos essa variável mais tarde enquanto fazemos os códigos.

Você irá encontrar o bloco de código `10 createEdgeSprites();` na seção

Sprites

4. **Escrever a função:** Agora vamos escrever o código dentro da função 'draw'.

P: Por quê escrevemos o código dentro da função 'draw'?

R: O código dentro da função 'draw' é executado continuamente. Assim, os carros continuarão descendo e, então, reaparecendo no topo continuamente, criando uma ilusão de efeito de condução para o jogador.

Escreva o seguinte código dentro da função 'draw':

- Confira se a tecla da seta **esquerda** no teclado está pressionada usando a 'condição **if**'. Então, a condição deve ser - se a tecla da seta **esquerda** estiver pressionada, precisamos mover o carro do jogador para a esquerda. Assim, adicione a 'condição **if**' dentro da função **draw**.

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

Então, dentro dos parênteses (), insira esse bloco de código

`keyDown (code)`

Esse bloco de código está disponível na seção

World

Este bloco de código `keyDown (code)` confere se uma tecla específica está pressionada ou não.

Se a tecla que estamos procurando for a seta **esquerda**, então o código dentro da função **keyDown()** deve ser definido como **esquerda (left)**.

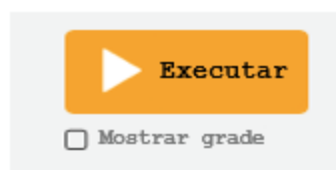
`keyDown (▼ "left")`

Desta forma, a 'condição **if**' deve ficar assim:

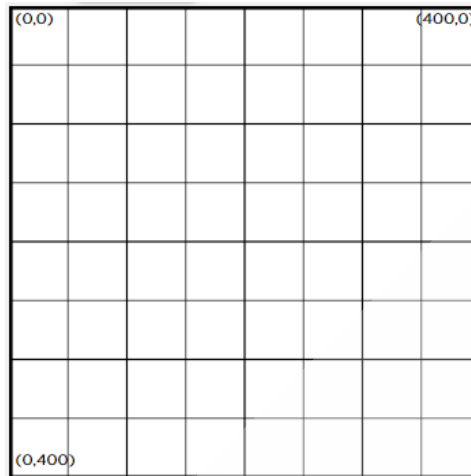
```
function draw() {
  drawSprites();
  if ( keyDown (▼ "left") )
  {
  }
}
```

E dentro desse bloco **if**, precisamos definir o valor da velocidade na direção X para **-3**. Agora, vamos entender o porquê do valor **-3**. Abaixo está a grade do Laboratório de Jogos.

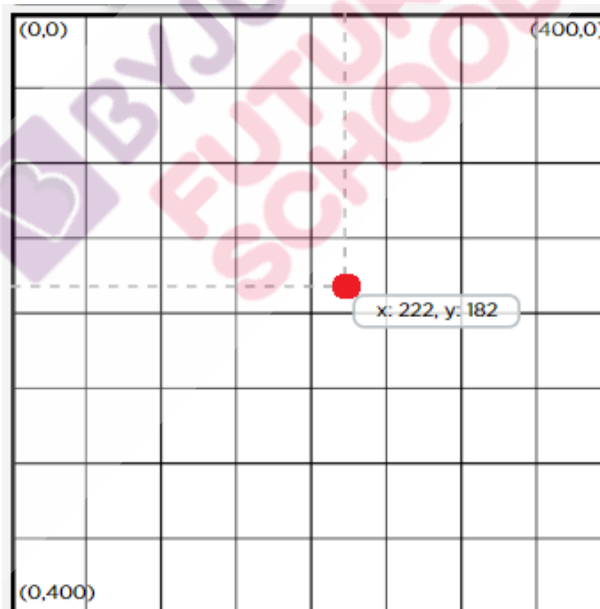
OBSERVAÇÃO: para ativar ou desativar essa grade, use a caixa que aparece abaixo do botão 'Executar'.



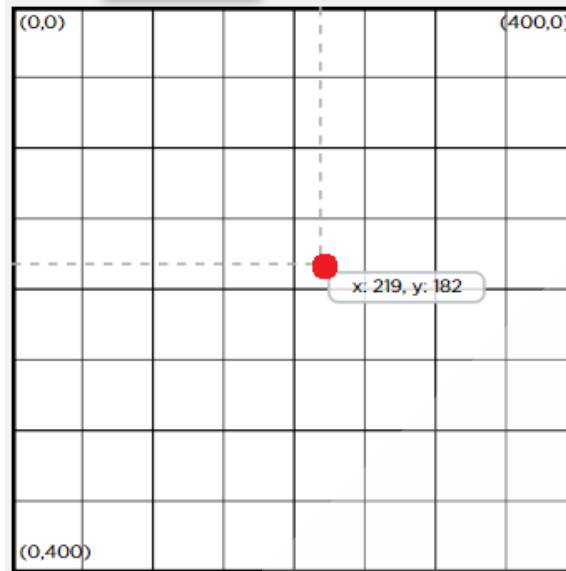
Uma vez que a opção 'Mostrar grade' estiver ativada, a tela ficará assim:



Supondo que estamos na posição $x = 222$ e $y = 182$



e queremos nos mover para a esquerda. Logo, precisamos reduzir o valor do eixo X. Mas isso depende de quanto queremos nos mover. Se quisermos andar 3 pixels para a esquerda, então precisamos reduzir 3 pixels no valor do eixo X. Como resultado, teremos $222 - 3 = 219$.



Assim, ao subtrair **3** da velocidade de X, o sprite pode se mover para a esquerda.

Primeiramente, arraste e solte esse bloco de código `sprite.velocityX`, e então altere seu nome para **carro_do_jogador**, pois queremos mover nosso carro do jogador e definir o valor para **-3**. Defina a velocidade de X para **-3**.

```
carro_do_jogador.velocityX = -3;
```

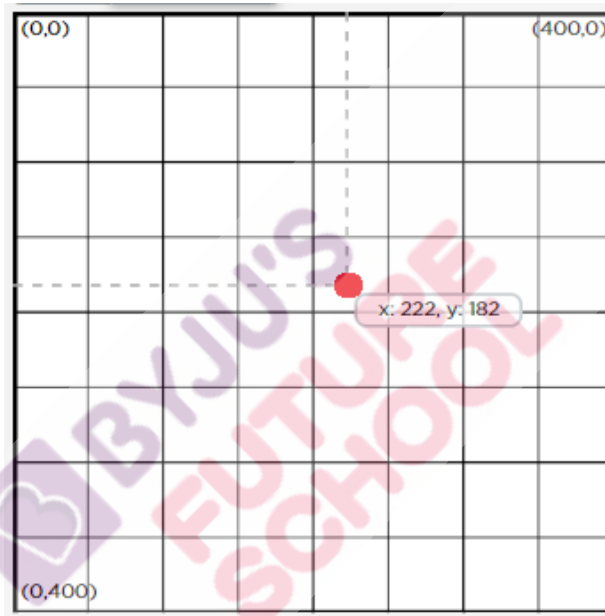
Aqui, o **carro_do_jogador** é o carro preto e ele vai se mover para a esquerda se pressionarmos a tecla da seta esquerda no teclado.

```
function draw() {
  drawSprites();
  if (keyDown(↵"left")) {
    carro_do_jogador.velocityX = -3;
  }
}
```

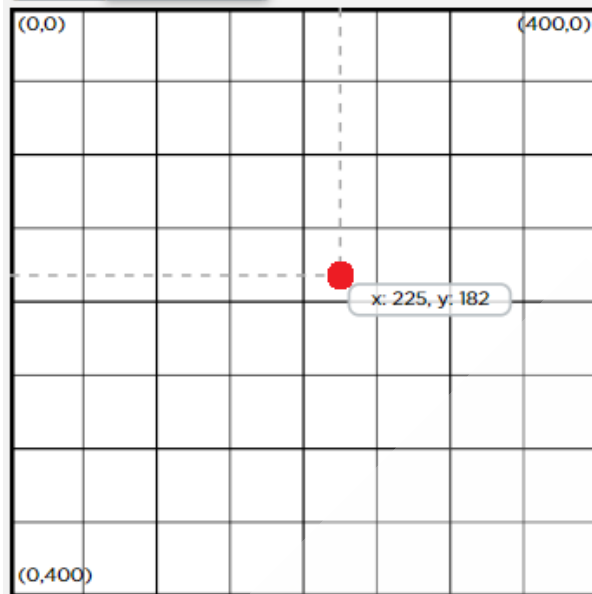

- b. Da mesma forma, confira se a tecla da seta **direita (right)** está pressionada.

```
if ( keyDown (▼ "right") ) {
```

Supondo que estamos na posição $x = 222$ e $y = 182$



e queremos nos mover para a direita. Logo, precisamos aumentar o valor do eixo X. Mas isso depende do quanto queremos nos mover. Se quisermos andar 3 pixels para a direita, então precisamos adicionar 3 pixels no valor do eixo X. Como resultado, teremos $222 + 3 = 225$. Então, para conseguirmos nos mover 3 pixels para a direita, teremos que adicionar 3 à velocidade X.



Assim, ao adicionar **3** à velocidade de X, o sprite pode se mover para a **direita**.

```
function draw() {
  drawSprites();
  if (keyDown(▼ "left")) {
    carro_do_jogador.velocityX = -3;
  }
  if (keyDown(▼ "right")) {
    carro_do_jogador.velocityX = 3;
  }
}
```

- c. Se o **carro_do_jogador** atingir o **carro1**: queremos verificar se o **carro_do_jogador** está encostando no **carro1**. Caso esteja, o jogador perde o jogo. Assim, exibiremos a tela de reinicialização. E, para exibir essa tela, precisamos mudar o fundo de reinicialização visível para **verdadeiro**.
- Substitua o nome do sprite com a variável **carro_do_jogador** e defina o **carro1** como alvo.

```
carro_do_jogador.isTouching(carro1)
```

- Agora, precisamos posicionar essa condição dentro da “condição if”. Essa será a condição que verifica se o carro_do_jogador está tocando No carro1.

```
function draw() {  
  drawSprites();  
  if (keyDown(▼"right")) {  
    carro_do_jogador.velocityX = 3;  
  }  
  if (keyDown(▼"left")) {  
    carro_do_jogador.velocityX = -3;  
  }  
  if (carro_do_jogador.isTouching(carro1)) {
```

Então, dentro da ‘condição if’, iremos escrever o código que torna o sprite do fundo de reinicialização visível quando o jogador perder o jogo.

Quando criamos o sprite de reinicialização, tivemos que definir sua visibilidade como **falsa** já que não queríamos que ele fosse exibido o tempo todo. Só queremos que ele apareça quando o jogador bater seu carro em qualquer outro carro inimigo. Mas agora, já que queremos que o sprite fique visível, vamos definir a visibilidade como **verdadeira**.

```
function draw() {
  drawSprites();
  if (keyDown("right")) {
    carro_do_jogador.velocityX = 3;
  }
  if (keyDown("left")) {
    carro_do_jogador.velocityX = -3;
  }
  if (carro_do_jogador.isTouching(carro1)) {
    recomeçar.visible = verdadeiro;
  }
}
```

- d. **Se o carro_do_jogador atingir o carro2:** agora, queremos verificar se o carro_do_jogador está encostando no carro2. Caso esteja, o jogador perde o jogo. Assim, exibiremos a tela de reinicialização. E, para exibir essa tela, precisamos mudar o fundo de reinicialização visível para **verdadeiro**.

- Substitua o nome do sprite com a variável **carro_do_jogador** e defina o carro2 como alvo.

```
carro_do_jogador.isTouching(carro2)
```

- Agora, precisamos posicionar essa condição dentro da “condição if”. Essa será a condição que verifica se o carro_do_jogador está tocando no carro2.

```
function draw() {
  drawSprites();
  if (keyDown(↴"right")) {
    carro_do_jogador.velocityX = 3;
  }
  if (keyDown(↴"left")) {
    carro_do_jogador.velocityX = -3;
  }
  if (carro_do_jogador.isTouching(carro1)) {
    recomeçar.visible = verdadeiro;
  }
}
```

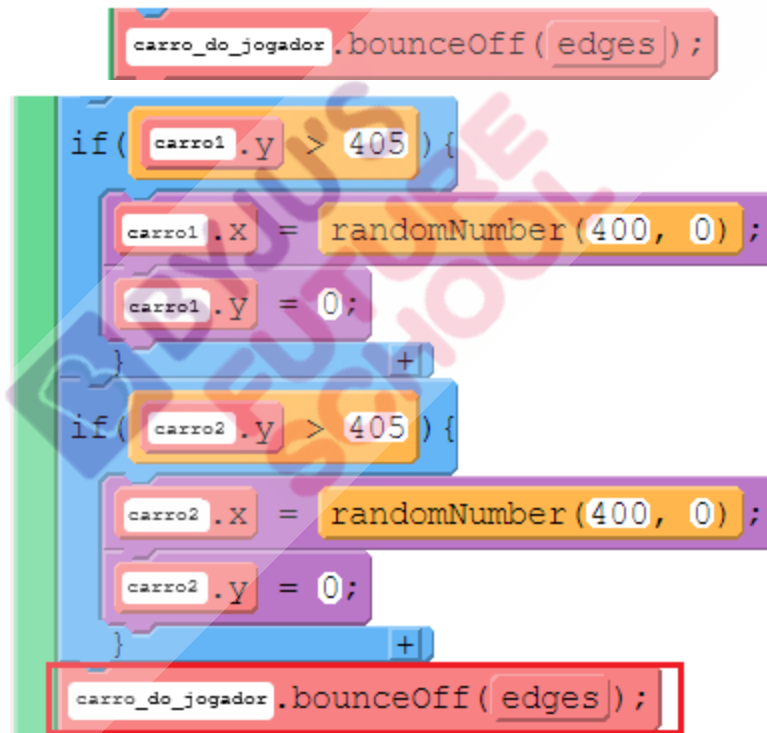
Então, dentro da 'condição if', iremos escrever o código que torna o sprite do fundo de reinicialização visível quando o jogador perder o jogo. Vamos definir a **visibilidade** como **verdadeira**.

```
if (carro_do_jogador.isTouching(carro1)) {
  recomeçar.visible = verdadeiro;
}
if (carro_do_jogador.isTouching(carro2)) {
  recomeçar.visible = verdadeiro;
}
```

- e. **Fazer o sprite quicar quanto tocar nas bordas:** agora, iremos escrever o código dentro da função **draw()** para fazer os sprites voltarem quando tocarem nas bordas. Então, precisamos definir que essa ação ocorra quando o sprite (carro

do jogador) tocar nas bordas da tela do jogo. Como queremos fazer nosso sprite quicar toda vez que tocar nos limites da tela, precisamos usar o bloco de código `sprite.bounceOff(target)`, que pode ser encontrado na seção **Sprites**.

- Substitua o nome do sprite para **carro_do_jogador**, uma vez que estamos configurando esta ação para o carro_do_jogador (o carro preto).
- Então, digite **bordas** dentro da função **bounceOff** e aperte **Enter**.



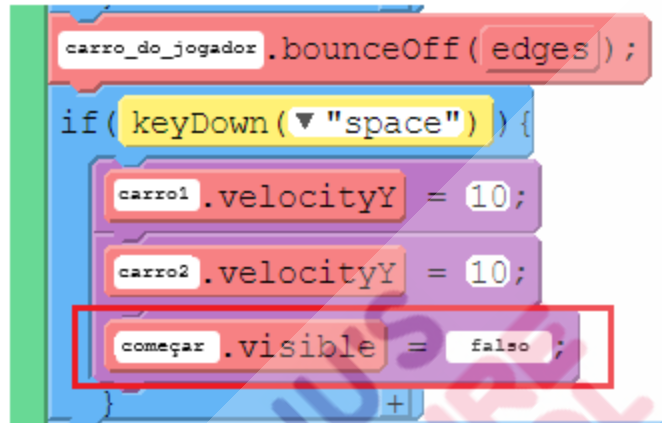
5. Visibilidade do fundo de inicialização: se você se lembra da demonstração da última aula, lembre que o jogo não começou imediatamente depois que o botão



foi pressionado. Primeiramente, mostramos uma imagem com as instruções para começar o jogo e como o jogador pode jogá-lo. Essa imagem está armazenada em uma animação chamada **começar**, para a qual nós já criamos um sprite no começo da aula. Como você viu, não escondemos esse sprite, por isso, se

 Executar

you press the button, you won't be able to play. Instead of this, you will see this background. Now, when the space bar is pressed, we want the game to start. Therefore, we will hide this background by defining its visibility as **false**. For this, inside the 'condition if', which is where we can check if the space bar is pressed, we will write the code to define the visibility of this background as **false**.



6. Recomeçar o jogo: we want the player to be able to start the game again if he loses. This can be done by clicking the **Reiniciar** button and, next, the button

 Executar

. But, as we are creating a complete application, we want the user to be able to restart the game in the application itself. Therefore, we will restart the game if the player clicks on the exit screen with the mouse. Therefore, we will check if the mouse button was pressed or not. If the mouse button was pressed, we will define the visibility of the background of the start of the game as **false** so that the game starts again.

→ For this, we will use the code block

`mousePressedOver(sprite)`

from the section

World

→ Replace the text **sprite** with the variable **recomeçar**, one time that the variable controls the sprite of the background of initialization and we want to check if the mouse is pressed over the sprite of the background of initialization.

`mousePressedOver(recomeçar)`

→ Then, position this code block inside the 'condition if', like this:

```

if ( keyDown(▼"space") ) {
    carro1.velocityY = 10;
    carro2.velocityY = 10;
    começar.visible = falso ;
}
+
if ( mousePressedOver ( recomeçar ) ) {

```


→ Agora, dentro da 'condição if', defina a visibilidade do sprite do fundo de reinicialização como **falso**, assim o fundo será escondido e o jogador poderá jogar o jogo novamente.




```

if ( keyDown(▼"space") ) {
    carro1.velocityY = 10;
    carro2.velocityY = 10;
    começar.visible = falso ;
}
+
if ( mousePressedOver ( recomeçar ) ) {
    recomeçar.visible = falso ;
}
+

```

Passo 4: Testar o jogo

Então, o 4º passo é executar o jogo para testá-lo. Vou clicar no botão  para testar o jogo. Assim que pressionar a **barra de espaço**, o jogo começa e os carros azuis e verdes se movem continuamente para baixo e reaparecem novamente no topo.

A Professora Para de Compartilhar a Tela	
<p align="center">  ATIVIDADE DO ALUNO - 30 min </p>	
<p>Agora é sua vez. Por favor, compartilhe sua tela comigo.</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 modo de Tela Cheia 	
O Aluno Inicia o Compartilhamento de Tela	
Dizer 	Fazer 
<p>Ok, agora é sua vez de começar a construir a mesma atividade.</p> <p>Abra a Atividade-do-Aluno-1, faça o login no code.org e continue a atividade com os códigos da última aula.</p> <p>Você pode me dizer os passos que vai seguir?</p> <p>Passo 1: Definir um propósito Neste passo, especificamos o que queremos que o jogo faça.</p> <p>Step 2: Fazer o design do jogo Neste passo, especificamos os componentes de design que queremos em nosso jogo.</p>	<p>Atividade do Aluno 1-CODE.ORG</p> <p>Atividade do Aluno 2-RECURSOS.</p> <p>Se o aluno não tiver o código feito na aula anterior, forneça ao aluno o link mencionado na Atividade-da-professora-4 e peça-lhe para remixar e começar a programar.</p>

Passo 3: Programar as regras:

Neste passo, nós especificamos as regras do jogo e o seu respectivo código.

Passo 4: Executar o jogo:

Neste passo, nós executamos o jogo e verificamos se ele está funcionando.

**OBSERVAÇÃO: peça que o aluno faça o seguinte:**

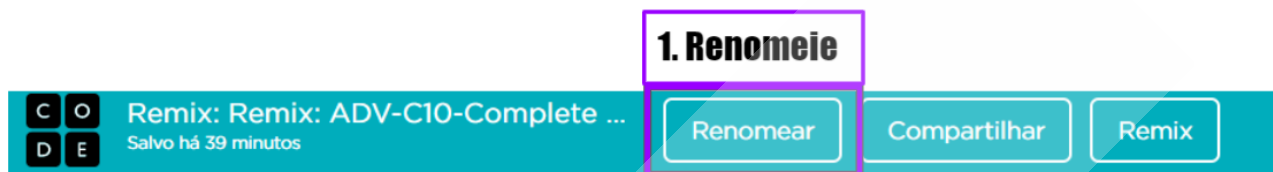
1. Crie animações para o carro do jogador, para o fundo que é mostrado antes de iniciar o jogo e para o fundo que é mostrado quando o jogador perde o jogo, que é o fundo de reinicialização. Baixe as imagens da [Atividade-do-aluno-2](#).
2. Programar para criar o sprite do carro do jogador e ambos os sprites de fundo, e definir as animações para todos os 3 sprites.
3. Dentro da função draw, siga os passos abaixo:
 - a. Programar para verificar se a tecla de seta para a esquerda está pressionada. Então, dentro do bloco IF, escreva o código para mover o carro do jogador para a esquerda
 - b. Código para verificar se a tecla de seta para a direita está pressionada. Então, dentro do bloco IF, escreva o código para mover o carro do jogador para a direita.
 - c. Código para verificar se o carro do jogador está tocando no carro1. Então, dentro do bloco IF, escreva o código para exibir o sprite de fundo de reinicialização, já que o jogador perdeu o jogo.
 - d. Código para verificar se o carro do jogador está tocando no carro2. Então, dentro do bloco IF, escreva o código para exibir o sprite de fundo de reinicialização, já que o jogador perdeu o jogo.
 - e. Código para quicar no carro do jogador se o carro do jogador tocar em qualquer uma das

- bordas da tela.
- f. Código para quicar no carro do jogador se o carro do jogador tocar em qualquer uma das bordas da tela..

Ótimo, excelente!

Procedimento para compartilhar

1. Clique no botão **Renomear**.



2. Escreva o nome do projeto



3. Clique em **Salvar** para salvar o nome do projeto.
4. Clique em **Compartilhar** para compartilhar o jogo com seus amigos.



TESTE BETA DO APLICATIVO

Ótimo!

Portanto, sua tarefa será fazer um teste BETA de seu aplicativo de jogo, compartilhando seu aplicativo com pelo menos 5 amigos e obtendo feedback sobre se eles gostaram ou não do aplicativo de jogo.


Você pode pedir aos seus pais para enviarem o link do aplicativo do jogo que você

gerou para os pais dos seus amigos no WhatsApp ou por e-mail para obter seus comentários.

Vou perguntar sobre o feedback de seus amigos na próxima aula e tentaremos trabalhar nisso juntos.

Professora Orienta o Aluno a Parar de Compartilhar a Tela

SESSÃO DE FECHAMENTO - 5 min

A professora inicia a apresentação de slides  dos slides 20 e 21.
Consulte as anotações da oradora e siga as instruções em cada slide.

DICAS E SUGESTÕES DO PROJETO - 5 min

A professora inicia apresentação de slides  do slide 22.

A Professora Inicia o Compartilhamento de Tela

Dizer



Fazer



FATIADOR DE FRUTAS

Objetivo do Projeto:

Na aula 16, você completou o jogo de corrida de carros definindo ações para o carro do jogador, como mover para a esquerda e para a direita, e então exibiu a imagem de fundo do reinício quando o carro do jogador toca em outro carro.

Neste projeto, você terá que praticar e aplicar o que aprendeu na aula para fazer um aplicativo para tocar em diferentes instrumentos musicais ao passar o mouse sobre os instrumentos.

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

História:

Sia adora frutas e vegetais. Então, a mãe de Sia pede ao seu filho para construir um jogo de fatiador de frutas para a Sia, de forma a aumentar o seu interesse em alimentos saudáveis e levar a hábitos alimentares saudáveis para Sia e seu irmão.

Você ajudará o irmão de Sia a construir este jogo?

Estou muito animada para ver a solução do seu projeto e sei que você se sairá muito bem.

Isso é tudo por hoje. Te vejo na próxima aula.

Até breve!

A Professora Clica em

× Terminar Aula

ATIVIDADE ADICIONAL DA PROFESSORA



A Professora Inicia o Compartilhamento de Tela

A professora inicia a apresentação de slides



dos slides 23 ao 27.

ATIVIDADES ADICIONAIS DO ALUNO



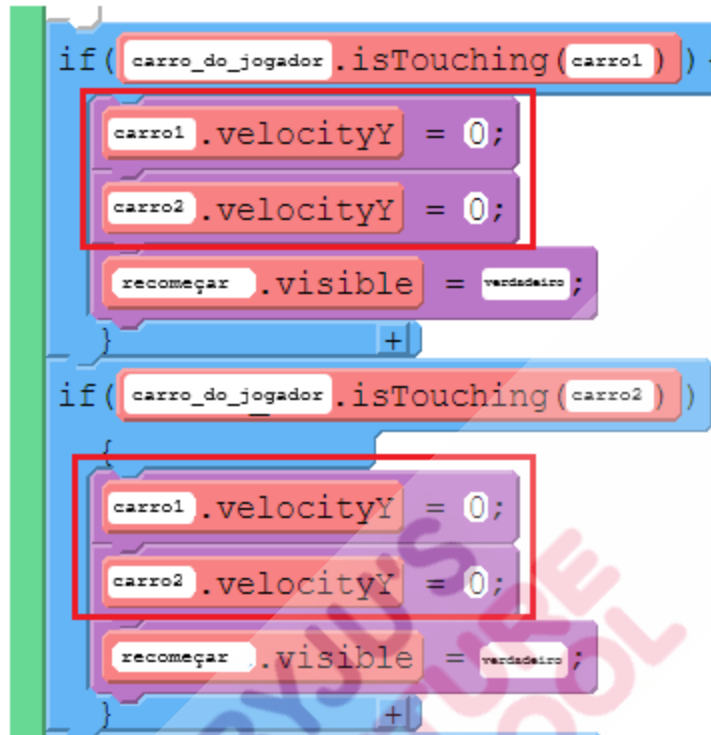
O Aluno Inicia o Compartilhamento de Tela

Atividades Adicionais:

- 1. AA1 - Impeça que todos os carros se movam quando o jogador tocar em outro carro:**

Quando o carro do jogador tocar em qualquer outro carro, o jogador perde o jogo e exibimos um sprite de fundo de reinicialização. Mas os carros não param de se mover.

Então, nesta atividade adicional, vamos parar os carros definindo sua velocidade para 0.



- 2. AA2 - Adicione um som de batida de carro quando o carro do jogador tocar em outro carro:**

```

if ( carro_do_jogador .isTouching ( carro1 ) ) {
  playSound(▼"sound://category_hits/8bit_splat.mp3", ▼false);
  carro1.velocityY = 0;
  carro2.velocityY = 0;
  recomeçar.visible = true;
}
if ( carro_do_jogador .isTouching ( verdadeiro ) ) {
  playSound(▼"sound://category_hits/8bit_splat.mp3", ▼false);
  carro1.velocityY = 0;
  carro2.velocityY = 0;
  recomeçar.visible = verdadeiro;
}

```

Atividade	Nome da Atividade	Links
Atividade da Professora 1	CÓDIGO COMPLETO	https://studio.code.org/projects/gamelab/jjNqka0JuE0paHcfA-tggseyABDuAawQX8C9ztfq6ZU
Atividade da Professora 2	PROGRAMANDO O JOGO	https://studio.code.org/projects/gamelab/qSk2NJ4Sx2FisWK0fEZSFzvzQuyJRoDznT5SC493aWQ
Atividade da Professora 3	RECURSOS	https://drive.google.com/drive/folders/1GtJ_BHcw4AJDBtSyD-cEGpXK_U_XTGh2?usp=sharing
Atividade da Professora 4	BACKUP - CÓDIGO FEITO NA AULA C15	https://studio.code.org/projects/gamelab/qSk2NJ4Sx2FisWK0fEZSFxxEJAI9M5ezuCFc96QnRQ Caso o aluno não consiga encontrar a atividade de corrida de carros feita na aula anterior, envie esse link a ele.

Atividade do Aluno 1	CODE.ORG	https://studio.code.org/
Atividade do Aluno 2	RECURSOS	https://drive.google.com/drive/folders/1GtJ_BHcw4AJDBtSyD-cEGpXK_U_XTGh2?usp=sharing
Resolução do projeto	FATIADOR DE FRUTAS	https://studio.code.org/projects/gamelab/J3H7GQ9hKmrHuRZ4cjjqnnjsTzV5LclC6RM9uFXj3vg
Link do recurso visual para referência da professora	Link do recurso visual	https://s3-whjr-curriculum-uploads.whjr.online/a3c39541-6323-45e3-852a-bf79a1958341.html
Referência da professora: Questionário de aula	Questionário de Aula	https://s3-whjr-curriculum-uploads.whjr.online/65a854e0-48a3-4623-a0f8-7f85bd1f059c.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/9cac0115-3080-41b5-8d50-8304dd956199.html