

Tópico	CONDICIONAIS	
Descrição da aula	As crianças aprendem a aplicar condições na lógica do programa para tornar um jogo interativo, emocionante e sem fim, aumentando sua confiança e habilidades de resolução de problemas.	
Aula	ADV-C15	
Duração	55 min	
Objetivo	<ul style="list-style-type: none"> • Comece a construir um jogo de corrida de carros que não tem fim. 	
Recursos necessários	<ul style="list-style-type: none"> • Recursos da Professora: <ul style="list-style-type: none"> ○ Laptop/PC com WebCam ○ Use suas credenciais de login Code.org ○ Fone de ouvido com microfone ○ Bloco de notas e caneta • Recursos do Aluno: <ul style="list-style-type: none"> ○ Laptop/PC com WebCam ○ Use suas credenciais de login Code.org ○ 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
<p>OBSERVAÇÃO: Se o aluno tiver um desempenho melhor do que o esperado, a aula deve ser ministrada na proporção de 50:50. 50% do código deve ser feito pelo aluno usando os blocos de código. E os 50% restantes do código devem ser feitos pelo aluno usando o MODO TEXTO.</p> <p>Habilitando o Modo-Texto: Clique no botão Mostrar Texto no canto superior direito da tela do code.org.</p>		

A única diferença entre o **Modo-Texto** e a programação **Baseada-em-Blocos** é -
A programação **Baseada-em-Texto** é feita escrevendo o código e a programação **Baseada-em-Blocos** é feita arrastando e soltando os blocos de código na área de trabalho.

Todos os programadores do mundo usam a programação baseada em texto, e a programação baseada em texto é aceita universalmente. Após alguns módulos, mudaremos para uma programação completamente baseada em texto. Consequentemente, você será solicitado a iniciar a programação baseada em texto a partir de agora.

SESSÃO DE INTRODUÇÃO - 5 min

A professora inicia apresentação de slides dos slides 1 a 19.
Consulte as anotações da oradora e siga as instruções em cada slide.



ATIVIDADE DA PROFESSORA - 10 min



A Professora Inicia o Compartilhamento de Tela

Dizer



Fazer



Nesta aula, vamos começar a construir um jogo de corrida de carros sem fim.

Você está animado com isso?

Demonstre o jogo completo.
Execute a [Atividade da professora - 1](#).



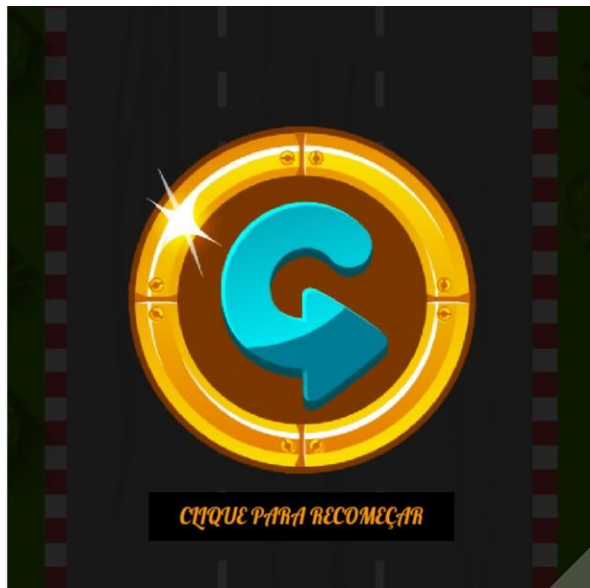
Pressione a **barra de espaço** para começar o jogo.



Assim que o jogo começar, pressione a tecla de seta para a **esquerda** e a tecla de seta para a **direita** para controlar seu carro e evitar ser atropelado pelos outros carros.

Se você for atropelado por outro carro, o jogo para e você

verá uma tela com a mensagem “CLIQUE PARA RECOMEÇAR”.

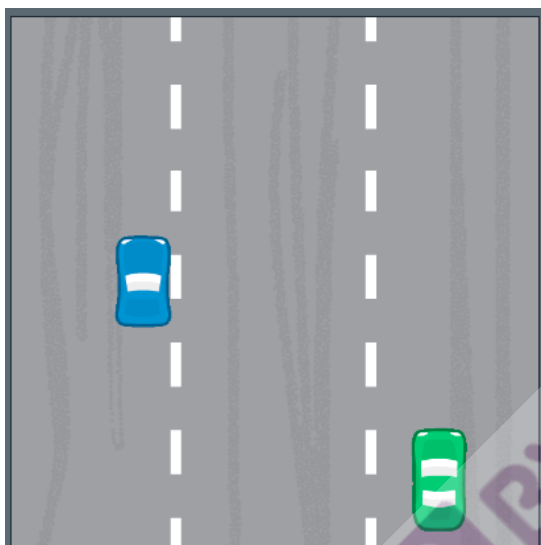


Se você clicar em qualquer lugar da tela do jogo, o jogo será reiniciado.

Como este jogo é grande, iremos completá-lo em duas aulas.

Demonstrar a saída proposta -

Pressione a barra de espaço para iniciar o jogo.



Nesta aula, criaremos os carros com um efeito de movimento para o jogo. Na demonstração, vimos como os carros inimigos estavam se movendo para baixo para criar um efeito do carro do jogador avançando. Criaremos esse efeito nesta aula. Os carros são colocados aleatoriamente no eixo x.

Excelente!

Agora que você entendeu o propósito do jogo, vamos começar a fazer o seu design.

Passo 2: Faça o design do jogo

Tendo em mente o propósito do jogo, o design do jogo exigiria os seguintes recursos:

1. Uma imagem inicial (que aparece quando iniciamos o jogo)
2. Uma imagem de fundo (a estrada)

3. Um carro inimigo verde
4. Um carro inimigo azul
5. Um carro de jogador
6. Uma imagem de 'Fim do jogo' - o fundo de reinicialização (que aparece se perdermos o jogo)

Adicionaremos apenas os recursos que são necessários para esta aula.

- Precisamos criar 3 animações:
 1. Um plano de fundo que é a estrada.
 2. Um carro inimigo verde
 3. Um carro inimigo azul



OBSERVAÇÃO: A professora deve completar o código que está na **Atividade da Professora - 3** enquanto explica as instruções abaixo. Na **Atividade da Professora - 3**, alguns dos códigos e animações são fornecidos previamente a você.

Abra a **Atividade da Professora - 3**, clique em



e então clique em



para que você possa editar o código. Comece explicando o código de acordo com as instruções fornecidas abaixo.

As etapas para criar uma nova animação são fornecidas a seguir:

Como já criei animações exceto para o segundo carro (carro verde), então começarei a criar a animação para o carro verde. Antes disso, vou procurar o carro verde nos recursos do Laboratório de Jogos e adicioná-lo à minha animação. Todos os recursos (imagem da estrada) são fornecidos na **Atividade da Professora - 4** para mim e, para você, na **Atividade do aluno - 2**.

Observação: peça ao aluno para fazer upload e adicionar a imagem da estrada ao criar a animação para a estrada.

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

Código Animação

2. Em seguida, clique em **Nova Animação**.



3. Agora, usando a barra de pesquisa, procure por carros e selecione qualquer uma das animações de carro que você deseja. Vou selecionar o carro verde.

Biblioteca de animação



4. Agora, altere o nome da animação para **carro2**.



A criação da nova animação está pronta.

Passo 3: Programar as regras

O código completo para hoje:

CÓDIGO BASEADOS EM BLOCOS -

```
var estrada = createSprite(200, 200);  
estrada.setAnimation(▼ "estrada");  
estrada.scale = 1;  
var carro1 = createSprite(100, 200);  
carro1.setAnimation(▼ "carro1");  
carro1.scale = 0.6;  
var carro2 = createSprite(325, 350);  
carro2.setAnimation(▼ "carro2");  
carro2.scale = 0.6;
```



```
function draw() {  
  drawSprites();  
  if( keyDown(▼"space") ) {  
    carro1.velocityY = 10;  
    carro2.velocityY = 10;  
  }  
  if( carro1.y > 405 ) {  
    carro1.x = randomNumber(400, 0);  
    carro1.y = 0;  
  }  
  if( carro2.y > 405 ) {  
    carro2.x = randomNumber(400, 0);  
    carro2.y = 0;  
  }  
  if( carro1.isTouching(carro2) ) {  
    carro1.x = randomNumber(400, 0);  
    carro1.y = 0;  
  }  
}
```

CÓDIGO BASEADO EM TEXTO -

```
var estrada = createSprite(200, 200);
estrada.setAnimation("estrada");
estrada.scale = 1;
var carro1 = createSprite(100, 200);
carro1.setAnimation("carro1");
carro1.scale = 0.6;
var carro2 = createSprite(325, 350);
carro2.setAnimation("carro2");
carro2.scale = 0.6;
function draw() {
  drawSprites();
  if(keyDown("space")){
    carro1.velocityY = 10;
    carro2.velocityY = 10;
  }
  if(carro1.y > 405){
    carro1.x = randomNumber(400, 0);
    carro1.y = 0;
  }
  if(carro2.y > 405){
    carro2.x = randomNumber(400, 0);
    carro2.y = 0;
  }
  if(carro1.isTouching(carro2)){
    carro1.x = randomNumber(400, 0);
    carro1.y = 0;
  }
}
```

Explicação:

1. **Crie o sprite:** primeiro, vamos criar todos os sprites necessários. Você pode ver que já criei sprites para criar o plano de fundo, que é uma estrada, e o carro azul. Então, eu só preciso criar um sprite para o carro verde e definir suas propriedades.

Observação: sempre que criar um sprite, a sequência em que você cria o sprite é muito importante. Por exemplo, se você criar primeiro os sprites de carro e depois adicionar o código para criar o sprite de estrada, o sprite da estrada será colocado sobre o sprite do carro. Dessa forma, o sprite do carro não será visível, pois o sprite da estrada cobre completamente a tela de saída.

OBSERVAÇÃO: portanto, peça ao aluno para primeiro criar o sprite da estrada e, em seguida, criar o carro.

Sequência em que os sprites devem ser criados:

1. Sprite da estrada
2. Sprites dos carros (azul e verde): qualquer um desses sprites pode ser colocado em seguida, não é necessário seguir nenhuma sequência para eles. Você pode colocar o carro azul ou o carro verde primeiro

a. Crie o sprite do carro verde:

```
var carro2 = createSprite(325, 350) → ;  
carro2.setAnimation(▼ "carro2");  
carro2.scale = 0.6;
```

Para isso, primeiro, vou arrastar e soltar este bloco

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

da seção

Sprites, e mudar o nome da variável de **sprite** para **carro2**.

```
var carro2 = createSprite(325, 350) → ;
```

Agora, defina $X = 325$ e $Y = 350$. Você pode definir o sprite para qualquer posição de sua escolha. Apenas certifique-se de que ele não esteja no centro, pois colocaremos o carro do jogador no centro da tela na próxima aula. E se o carro do jogador tocar em qualquer um dos outros sprites do carro no início do jogo, o jogador perderá e o jogo terminará.

b. Defina a imagem do carro verde:

Para isso, arraste e solte este bloco

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

da seção

Sprites

e altere o nome da variável de **sprite** para **carro2**. Além disso, defina a animação para **carro2**, pois carro2 representa o carro

verde.

```
carro2.setAnimation(▼ "carro2");
```

c. **Defina o tamanho do carro verde:**

Para definir o tamanho do sprite, use o bloco `sprite.scale`. 'Scale' significa definir o tamanho da imagem (dimensioná-la). O valor da escala pode ser qualquer número entre 0 e 1. Precisamos renomear a variável de **sprite** para **carro2** e definir a escala para 0,6, pois não queremos que o sprite seja muito grande.

```
carro2.scale = 0.6;
```

- 2. Escreva a função:** defina a função draw. Agora, vamos escrever o código dentro da função draw. Esta função será executada continuamente. Portanto, os carros continuarão descendo e reaparecerão no topo continuamente para criar um efeito de direção no carro do jogador.

```
function draw() {  
    
}
```

- a. Defina a função draw. Escrevemos o seguinte código dentro desta função draw.
- b. Use o bloco `drawSprites();` para posicionar/exibir todos os sprites na tela do jogo.
- c. Agora, vamos escrever uma condição **IF**. A lógica aqui é — se a barra de espaço for pressionada, queremos começar com o efeito de carro em movimento para os carros inimigos. Dessa forma, quando a barra de espaço for pressionada, definiremos a velocidade de Y para o **carro1** e para o **carro2**. Portanto, a condição será verificar se a barra de espaço está pressionada.

Adicione uma condição **if** dentro da função **draw**.

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

Então, dentro dos parênteses (), coloque este bloco de código

`keyDown (code)`

World

Você obterá esta forma na seção mundo

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

Como a chave que procurávamos é a **barra de espaço**, portanto, o código dentro da função **keyDown()** deve ser definido como **space**.

`keyDown (▼ "space")`

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

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

- Defina a velocidade de **Y** do **carro1** para 10. Esta é a velocidade na qual o sprite se moverá na direção **Y**. Usaremos este bloco

`sprite.velocityY`

que podemos pegar na seção

Sprites

Definirei a velocidade para 10. Você pode defini-la com qualquer valor de sua escolha. Quanto maior for o valor, mais rápido o carro inimigo se moverá para baixo e mais difícil será o jogo para o usuário.

```
function draw() {
  drawSprites();
  if (keyDown("space")) {
    carro1.velocityY = 10;
  }
}
```

- Da mesma forma, definiremos a velocidade de **Y** do **carro2** como 10. Esta é a velocidade na qual o sprite se moverá na direção **Y**.

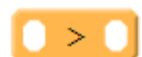
```
function draw() {
  drawSprites();
  if (keyDown("space")) {
    carro1.velocityY = 10;
    carro2.velocityY = 10;
  }
}
```

- d. Agora, vamos escrever outra condição **IF**. A lógica aqui é — se o carro1 for além de 405 pixels do eixo Y, o carro deve desaparecer, e queremos que ele reapareça de cima para criar a ilusão de mais e mais carros se movendo para baixo. Portanto, faremos o mesmo carro, que é o **carro1**, reaparecer de cima, definindo sua posição na parte superior da tela do jogo. Portanto, a condição será verificar se a posição Y do carro1 é maior do que 405 ou não.


```
if ( ) {
  +
}
```

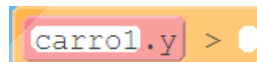
Então, adicione uma condição **if** dentro da função **draw**.

Então, dentro dos parênteses (), coloque este bloco de código

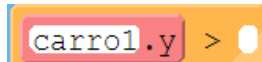


Você pegará este bloco na seção **Math**. Estamos usando este bloco porque temos que verificar se a posição y do **carro1** é maior que 405 ou não.

No lado esquerdo deste bloco , obteremos a posição y do carro1. Para isso, vamos usar o bloco **sprite.y** da seção **Sprites**. Este bloco de código fornece a posição y do sprite. Agora, substitua o **sprite** por **carro1**, pois queremos obter a posição do carro1.



Então, como queremos verificar se a posição do carro1 é maior que 405 ou não, então escreveremos 405 no lado direito do bloco



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

```
function draw() {
  drawSprites();
  if (keyDown("space")) {
    carro1.velocityY = 10;
    carro2.velocityY = 10;
  }
  if (carro1.y > 405) {
```

Dentro desse bloco de código, atualizaremos as posições x e y do carro1 para que ele reapareça novamente.

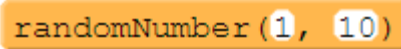
- **Defina uma posição x aleatória para o carro1:** usaremos o

bloco **sprite.x** da seção **Sprites**.

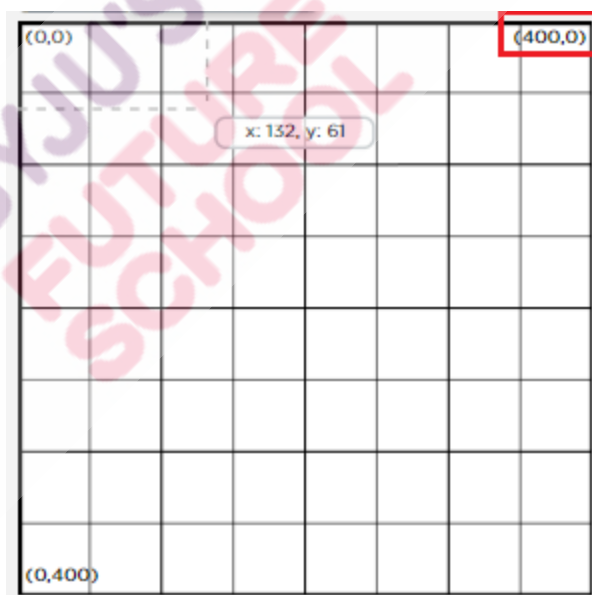
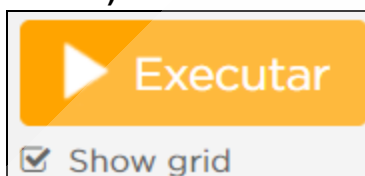
Vamos substituir o texto **sprite** por **carro1**, pois queremos atribuir uma posição x aleatória para o **carro1**.

E coloque um bloco de código para gerar números aleatórios no

lado direito do bloco .

Então, usaremos o bloco  da seção

Math para gerar números aleatórios. Verifique o número de pixels no eixo x da grade ativando a caixa de seleção **Mostrar grade (Show Grid)** abaixo do botão **Executar**.



Você pode ver que (400, 0) representa (x, y) — isso significa que estamos no pixel 400 do eixo x da tela de saída. Portanto, devemos gerar qualquer número aleatório entre 400 e 0.


```
if (carro1.y > 405) {
  carro1.x = randomNumber(400, 0);
}
```

Isso fará com que o carro seja colocado aleatoriamente em qualquer posição no eixo x (horizontalmente).

- **Defina a posição y para o carro1:** agora, definiremos a posição y do carro1 como 0, pois queremos que o carro reapareça no topo.

Portanto, vamos usar o bloco `sprite.y` da seção

Sprites

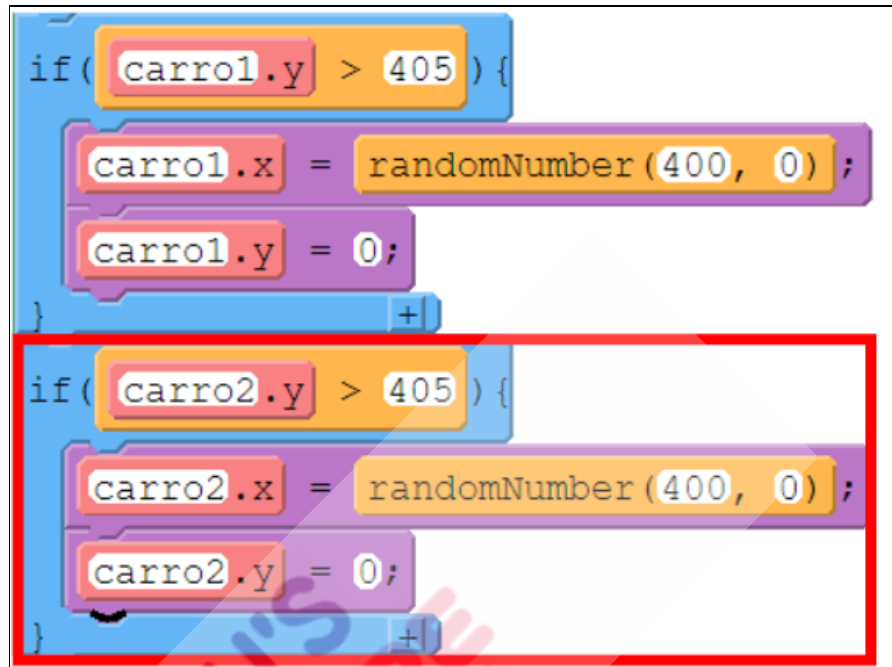
, pois queremos definir a posição y do carro1 como 0.

Portanto, coloque o valor **0** no lado direito do bloco

```
carro1.y = 0;
```

```
if (carro1.y > 405) {
  carro1.x = randomNumber(400, 0);
  carro1.y = 0;
}
```

Da mesma forma, verificaremos se a posição y do carro2 é maior que 405. Se sim, atualize a posição x e y do carro2 para que ele apareça em uma posição x aleatória a partir do topo.



- e. **Verifique se o carro1 toca no carro2:** queremos verificar se o carro1 está tocando no carro2. Caso esteja, precisamos fazer o carro1 desaparecer e reaparecer no topo, porque tocar em um carro enquanto dirige significa que ocorreu um acidente. Dessa forma, o carro1 deve ser destruído e, portanto, deve reaparecer no topo. Nós precisamos:

- I. Pegar o bloco **isTouching** da seção **sprite**, substituir o nome do sprite pela variável **carro1** e definir o alvo como **carro2**.

`carro1.isTouching(carro2)`

- II. Agora, precisamos colocar essa condição dentro da condicional **if**. Essa será a condição para verificar se o carro1 está tocando no carro2.


```
if (carro1.y > 405) {  
  carro1.x = randomNumber(400, 0);  
  carro1.y = 0;  
}  
if (carro2.y > 405) {  
  carro2.x = randomNumber(400, 0);  
  carro2.y = 0;  
}  
if (carro1.isTouching(carro2)) {
```

Então, dentro dessa condição if, vamos escrever o código para fazer o carro1 aparecer em uma posição aleatória no topo da tela de saída, assim como fizemos quando a posição y dos carros 1 e 2 era maior do que 405.

```
if (carro1.y > 405) {  
  carro1.x = randomNumber(400, 0);  
  carro1.y = 0;  
}  
if (carro2.y > 405) {  
  carro2.x = randomNumber(400, 0);  
  carro2.y = 0;  
}  
if (carro1.isTouching(carro2)) {  
  carro1.x = randomNumber(400, 0);  
  carro1.y = 0;  
}
```

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

ATIVIDADE DO ALUNO - 30 min

Agora é sua vez. Por favor, compartilhe sua tela comigo.

- 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



Ok, agora é sua vez de começar a construir a mesma atividade.

Você pode me dizer os passos que vai seguir?

Passo 1: Definir um propósito

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

Step 2: Fazer o design do jogo

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

[Atividade do Aluno 1 - CODE.ORG](#)

[Atividade do Aluno 2 - RECURSOS](#)

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.

Abra a [Atividade do Aluno - 1](#) e comece a programar a atividade.

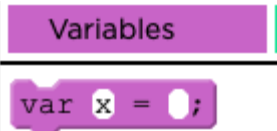
Você (o aluno) precisa fazer o seguinte:

1. Criar animações para o plano de fundo (a estrada) e para ambos os carros (verde e azul). Baixe as imagens da [Atividade do Aluno - 2](#).
2. Criar os sprites do plano de fundo, de ambos os sprites de carro e definir as animações para todos os 3 sprites.
3. Definir a função draw. Dentro da função draw, siga as seguintes etapas:
 - a. Adicionar a função drawSprite().
 - b. Código para verificar se a barra de espaço está pressionada. Então, dentro do bloco IF, escreva o código para mover os sprites carro1 e carro2 para baixo.
 - c. Código para verificar se a posição y do carro1 é maior que 405. Então, dentro do bloco IF, escreva o código para fazer o carro1 reaparecer em uma posição aleatória do topo.
 - d. Código para verificar se a posição y do carro2 é maior que 405. Então, dentro do bloco IF, escreva o código para fazer o carro2 reaparecer em uma posição aleatória do topo.
 - e. Código para verificar se o carro1 está tocando no carro2. Então, dentro do bloco IF, escreva o código para fazer o carro1

<p>reaparecer em uma posição aleatória do topo.</p> <p>Excelente! Maravilhoso!</p> <p>Mantenha este código em segurança, pois continuaremos programando e completaremos este jogo na próxima aula.</p>	
Professora Orienta o Aluno a Parar de Compartilhar a Tela	
SESSÃO DE FECHAMENTO - 5 min	
<p> A professora inicia a apresentação de slides dos slides 20 e 21. Consulte as anotações da oradora e siga as instruções em cada slide.</p>	
DICAS E SUGESTÕES DO PROJETO - 5 min	
<p> A professora inicia apresentação de slides do slide 22.</p>	
A Professora Inicia o Compartilhamento de Tela	
<p>Dizer</p> 	<p>Fazer</p> 
<p>BANDA DE MÚSICA</p> <p>Objetivo do projeto: Na aula 15, você aprendeu a criar animações de carros em movimento no Laboratório de Jogos.</p> <p>Neste projeto, você terá que praticar e aplicar o que aprendeu na aula para fazer um aplicativo para tocar diferentes instrumentos musicais ao passar o mouse sobre o instrumento.</p>	<p>Abra o link da Resolução do Projeto e faça uma demonstração do projeto para o aluno</p>

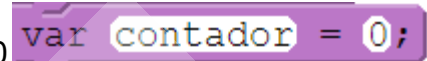
<p>História:</p> <p>Arte Musical é uma escola de música que ensina as crianças a tocar diversos instrumentos musicais e a identificar o som tocado pelos instrumentos. Então, eles estão procurando por alguém que possa criar um aplicativo que toque músicas diferentes de vários instrumentos musicais para que os alunos aprendam e entendam os diferentes sons produzidos por diferentes instrumentos musicais.</p> <p>Você vai ajudar os alunos da Arte Musical?</p> <p>Estou muito animada para ver a solução do seu projeto e sei que você se sairá muito bem.</p> <p>Isso é tudo por hoje. Te vejo na próxima aula.</p> <p>Até breve!</p>	
<p style="text-align: center;">A Professora Clica em </p>	
<p style="text-align: center;">Atividades Adicionais</p>	
<p style="text-align: center;">A Professora Inicia o Compartilhamento de Tela</p>	
<p style="text-align: center;">A professora inicia apresentação de slides  dos slides 23 a 26.</p>	
<p style="text-align: center;">ATIVIDADES ADICIONAIS DO ALUNO </p>	
<p style="text-align: center;">O Aluno Inicia o Compartilhamento de Tela</p>	
<p>Atividades Adicionais:</p>	

1. Vamos contar quantos carros já passaram. Para isso, primeiro, pegue o bloco de

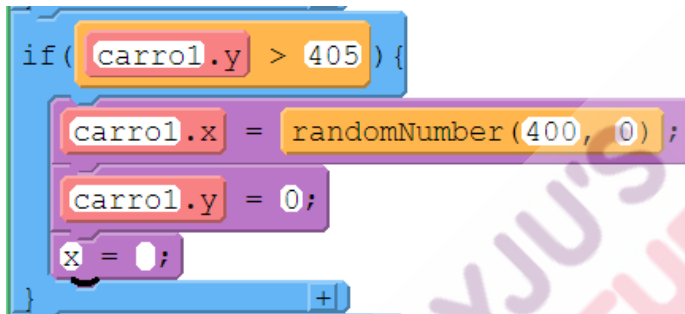


código de uma variável e arraste e solte-o fora da função **draw ()**.

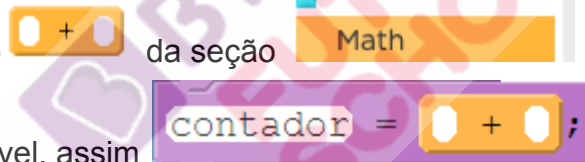
2. Em seguida, altere seu nome e defina seu valor como 0



3. Agora, arraste e solte este bloco de código dentro da condição if em que estamos verificando se o valor do eixo y do carro1 é maior que 405.



4. Pegue este bloco da seção Math e o coloque no lado direito



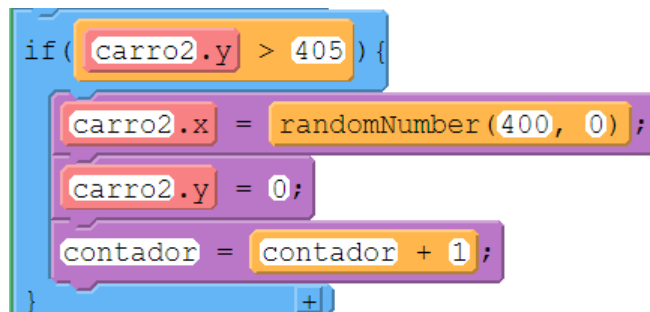
do bloco da variável, assim

5. Agora, do lado esquerdo deste bloco, escreva contador (como o nome da sua variável) e, no lado direito, escreva 1 (porque precisamos acrescentar 1 a cada

contagem de carro):



6. Faça o mesmo dentro da condição if onde estamos verificando se o valor do eixo y do



carro2 é maior que 405, assim - }

7. Agora, vamos obter o bloco de código de texto `text(str, x, y)` da seção `Drawing` e coloque no final da função `draw()`.
8. E substitua `str` pela variável que é a contagem e defina os valores dos eixos `x` e `y`:
`text(contador, 150, 30);`
9. Agora teste o código. E se você quiser aumentar o tamanho do texto, use este bloco `textSize(25);` da seção `Drawing`.

Atividade	Nome da atividade	Links
Atividade da Professora 1	CÓDIGO COMPLETO	https://studio.code.org/projects/gamelab/jjNqka0JuE0paHcfA-tggseyABDuAawQX8C9ztfq6ZU
Atividade da Professora 2	DEMONSTRAÇÃO O C15	https://studio.code.org/projects/gamelab/V7OX_f8uEFrjWqrq33yAKcJehzys7w0asBKRGFHmSs
Atividade da Professora 3	COMPLETE O CÓDIGO	https://studio.code.org/projects/gamelab/V7OX_f8uEFrjWqrq33yAKUL3jqmSwLV-IZCWNCFKD8
Atividade da Professora 4	RECURSOS	https://drive.google.com/drive/folders/1NRnbp2xXBe2FtXENnotf5Ubzgg15dMU-
Atividade do Aluno 1	CODE.ORG	https://studio.code.org/projects/gamelab/ggruFo7bXs1MsN-GS21hdsPh4pRwJ84awE4eNrTd3ig/edit
Atividade do Aluno 2	RECURSOS	https://drive.google.com/drive/folders/1op43EiwXUMg96BZY4M7XRSmvN8UFxnO?usp=sharing
Resolução do projeto	BANDA DE MÚSICA	https://studio.code.org/projects/gamelab/BJ-pRd42nbIBtsl-8FcdccocFA6QvVGMIAzjFfmbPA8
Link do recurso visual para referência da professora	Link do recurso visual	https://s3-whjr-curriculum-uploads.whjr.online/a7a0718c-b206-41f3-b917-27c03eed2bb1.html
Referência da professora: Questionário de aula	Questionário de Aula	https://s3-whjr-curriculum-uploads.whjr.online/3db93b55-6b57-404c-8a84-4cdc094de1dd.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/872be082-727e-46a8-b06a-f48bbd4fbf12.html

