



UMA INICIATIVA DE



With the participation  
of UNESCO



# AFRICA CODE WEEK

## Introduzindo a linguagem de programação Scratch

por

Brendan Smith, Camden Education Trust, Irlanda.  
TRADUZIDO POR INTERRMUN'Ã



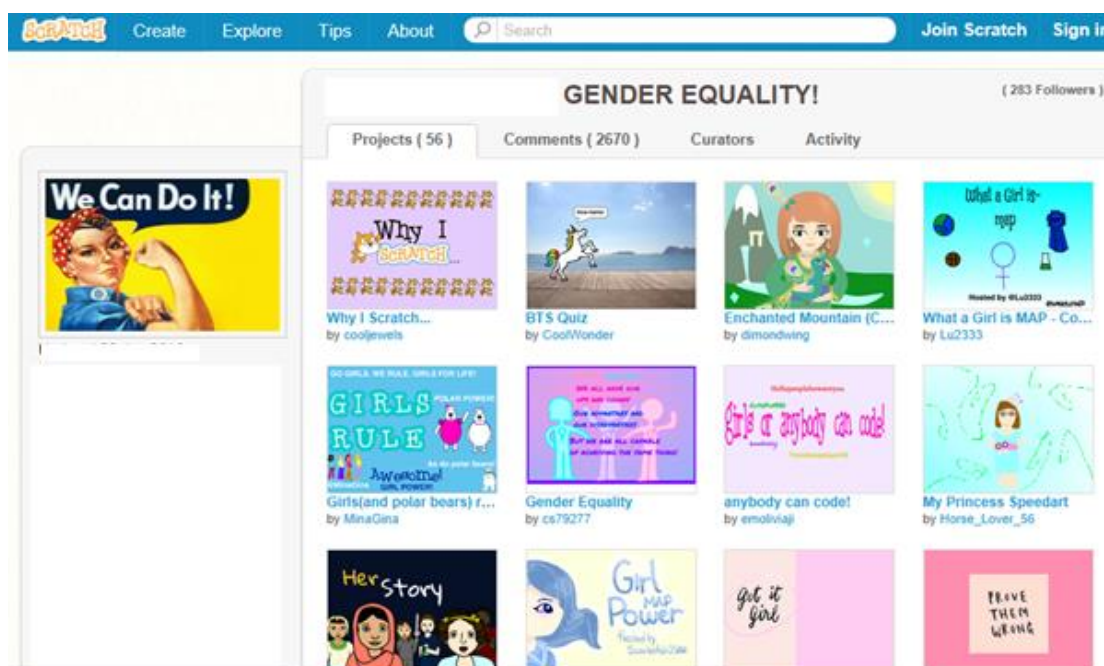
Africa Code Week é uma iniciativa da SAP. O conteúdo inteiro desenvolvido por Brendan Smith a favor do Camden Education Trust, é propriedade intelectual da Africa Code Week.

# Sumário

Introduzindo Scratch e a arte da codificação .....	3
Lição 1 - Introdução à interface Scratch.....	6
Lição 2- Codificação : as primeiras etapas .....	8
Lição 3 - Inserir sons em guião.....	16
Lição 4 – Melhorar a marcha do actor.....	19
Lição 5 - Actores multicoloridos.....	21
Lição 6 - Modificar o palco .....	23
Lição 7 – Um cão que late e persegue o gato ! .....	31
Lição 8 – Criar um actor.....	36
Lição 9 – Os actores controlados por cursor .....	39
Lição 10 – O actor psicodélico .....	40
Lição 11 - Interacção do actor .....	41
Lição 12 - Dois actores estão a conversar .....	44
Lição 13 – Criar um recife de coral .....	46
Lição 14 – A bola-alvo .....	51
Lição 15 - Os actores dançantes.....	54
Lição 16 - Desenhar formas .....	57
Lição 17 – A bola que ressalta.....	63
Lição 18 – Desenhar à mão livre.....	66
Lição 19 – Levar o cão a passear.....	68
Lição 20 – Planeamento e concepção dum jogo .....	76
Lição 21 - Jogo – Ataque de tubarão ! .....	77
Lição 22 - Ataque de tubarão avançado.....	84
Lição 23 – Jogo de ténis solitário .....	94
Lição 24 – Jogos de aventura: o labirinto incrível ! .....	98
Lição 25 – Caçador de demónios .....	105
Lição 26 - Extensão do jogo Caçador de demónios.....	109
Lição 27 – Jogo de atirar: os asteróides.....	114
Lição 28 – Jogos de dois jogadores.....	122
Lição 29 – Jogos de dois jogadores - ténis para dois .....	127
Lição 30 – Quizz de geografia : viajar por um continente .....	130

# Introduzindo Scratch e a arte da codificação

De que forma a codificação numérica pode agir como estreita aprendizagem interdisciplinar



Neste trecho, vamos apresentar-lhes o sistema Scratch, uma linguagem de programação desenvolvida no Media Lab do MIT nos Estados Unidos e que despertou a imaginação das crianças no mundo inteiro.

Scratch permite aos utilizadores uma criação fácil das suas próprias histórias e animados, de seus jogos, de sua música, de suas obras de arte interativas e o compartilhar destas criações pelo Web. Aparece ideal para as crianças com idade das oito às onze anos.

## O que é uma programação?

Uma programação é o pacote das instruções que indicam ao computador ou a qualquer outro aparelho eletrónico que tem de fazer. Estas instruções ou encomendas vêm sendo escritas em linguagem artificial (ou seja não falada). Várias vezes o guião utilizado é chamado código ou código numérico.

A programação numérica ou codificação é o processo de escrita do código.

## De onde vem o nome Scratch ?

O sistema Scratch, implementado por Media Lab do MIT nos Estados Unidos, é o nome que foi dado a esta linguagem de programação adaptada para as crianças ainda que potente.

O nome Scratch foi escolhido por seus inventores em razão do seu ponto comum com o método usado por um DJ de hip-hop que misturava diferentes músicas criando novos sons.



Pois um elemento-chave do site Scratch é que os membros podem descarregar os jogos numéricos de outras pessoas e se desejarem modificar os pormenores (por exemplo: « acelerar » ou « retardar » as personagens animadas no jogo). Como resultado, grande número destes projectos on-line são produtos de remixas de outros projectos no site <http://scratch.mit.edu> .

### Uma forma divertida de aprender

A linguagem Scratch leva semelhanças com os jogos de combinação infantis com blocos. Usa uma estrutura simples de **blocos** gráficos de código numérico que se encaixam para criar e controlar o som, a música e as imagens. Desta forma, corresponde perfeitamente ao gosto e à aprendizagem das crianças, pois serve-se da tendência natural para a criação com materiais de construção para realizar diferentes formas, jogos e histórias no meio de uma **nova** dimensão numérica **interativa empolgante**. Acabou a época das programações numéricas com textos de sintaxe difícil e aborrecida.



### Benefícios educativos : cálculo, alfabetização e muito mais

Scratch providencia às crianças e aos mais jovens um ambiente para desenvolver e usar seus talentos artísticos e criativos pela criação de animações atractivas.

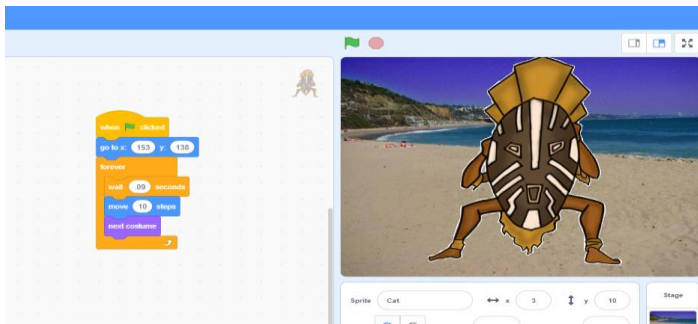
Durante as sessões de formação ,as crianças trabalham conjuntamente para realizar determinados

projectos, para aprender a planejar, para conceber e partilhar tarefas. São animados a mostrarem e a explicarem seus trabalhos acabados para os outros participantes. Ali a análise crítica e os elogios de seus colegas de turma faz parte do processo de aprendizagem de apresentação e de comunicação.

O sistema Scratch leva as crianças a despertarem suas competências artísticas e criativas num mundo numérico estimulador , gratificante e aventureiro ao mesmo tempo .

O ensino do Scratch usa muitos aspectos diferentes dos programas escolares do ensino primário como as artes, os idiomas e mais particularmente o cálculo e a alfabetização .

Como as crianças gostam de criar e de partilhar os projectos Scratch, elas aprendem importantes conceitos da matemática e da informática tais como a aritmética (soma, subtração, multiplicação, a divisão dos números), a geometria (um ramo da matemática relativo às medidas, à relação dos pontos, às linhas, dos



ângulos etc.) com termos tais como coordenadas, formas, tamanho, posição relativa das figuras), álgebra (dados variáveis, símbolos representando números para explicar quantidades e relações numéricas) tais como

noções adicionais tais como algoritmos, isso tudo ao aprender a raciocinar de forma sistemática, a resolver problemas, a trabalhar juntos e a desenvolver competências de apresentação..



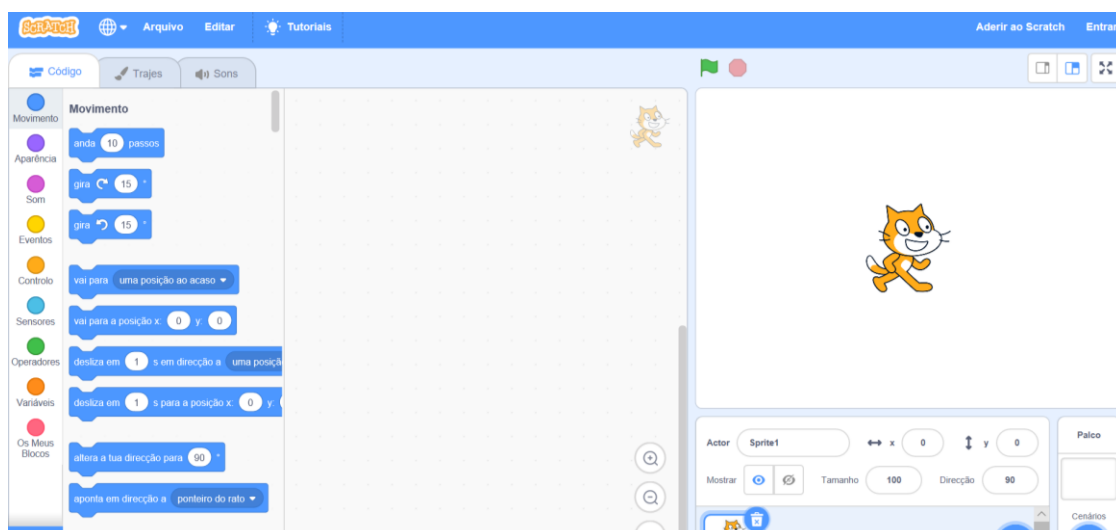
Muito raramente, para os alunos de Scratch, a motivação tem sido um problema.

Scratch dá a possibilidade aos utilizadores de criar uma grande variedade de projectos on-line que reflitam seus interesses pessoais e que possam ser usados como recursos em várias disciplinas escolares, por exemplo; criar jogos interativos, mapear uma

visita de diversos países em uma aula de geografia, produzir um concerto para uma aula de música, fabricar uma boneca com várias roupas numa aula de arte ou criar um relato numérico de um jogo desportivo, um conto folclórico ou estudar uma temática científica importante tal como as causas do aquecimento climático.

Com os alunos do primário, o docente pode ampliar os elementos artísticos de Scratch pedindo-lhes a fabricação de modelos dos seus actores feitos de barro quando planejam seus projectos.

# Lição 1 - Introdução à interface Scratch



## Partilha e armazenamento de projectos Scratch on-line

Vários projectos que foram usados nestas anotações de aprendizagem se encontram armazenados on-line site Web de Scratch.

Neste caso, está indicado o endereço Web específico do projecto individual.

Este site deveria ser usado também para armazenar os melhores exemplos de projectos criados por seus estudantes participantes, para que outros orientadores possam usá-los como recursos pedagógicos em suas aulas e como exemplos para outros participantes.

### Para acessar o site Scratch:

Digitem na barra de endereços: <http://scratch.mit.edu>

Caso ainda não tiverem conta, cliquem em **Aderir ao Scratch**

**Aderir ao Scratch** **Entrar**

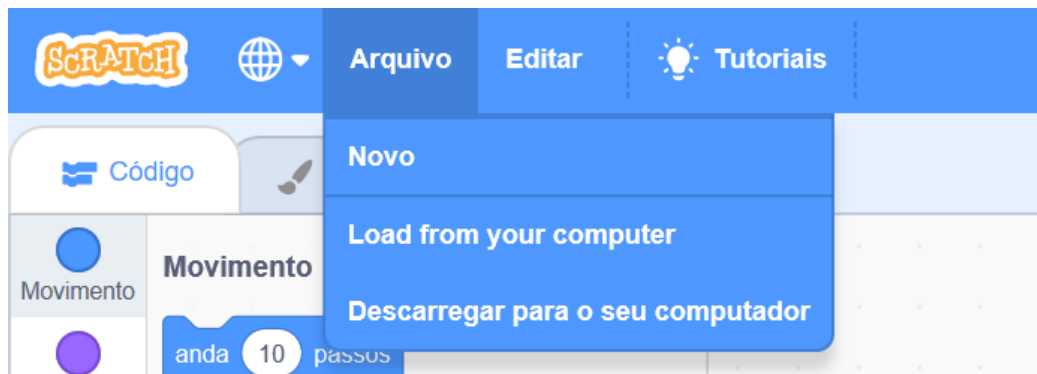
Mesmo assim, vocês podem aderir à comunidade existente « Code for Africa » ao ir à opção de conexão e

Debaixo de **Nome de Utilizador**, digitem « africacodeweek »

Debaixo de **Palavra-Passe**, digitem « fionnfionn ».

Para descarregar um projecto terminado desde seu computador ou outro dispositivo na conta Africa Code Week no site Scratch, primeiro acessem a opção Criar na barra de menus.

Logo debaixo da opção **Arquivo** seleccionem **Load from your computer**



Como docente, talvez tenha a vontade de criar uma conta no site Scratch para sua turma, sua escola ou para seu grupo e que possa ser usado para hospedar exemplos para cada tarefa de seu estudante.

Este curso usará então o **editor Scratch off-line**

Accessem um navegador Web tal como Google Chrome, Safari ou Firefox.

Digitem na barra de endereços :

<http://scratch.mit.edu/scratch3download/>

e sigam as instruções.

Acostumem-se com o programação , clicando em **Tutoriais**.

## Lição 2- Codificação : as primeiras etapas

Um Actor que fala e que se move

Um **Actor** é um personagem ou um objecto animado na sua programação.



Em Scratch, os Actores podem deslocarem-se, serem activos ou serem objectos imóveis. Vamos escolher um personagem actor a quem vamos dar vida.

Ainda assim, é importante saber que um actor não pode fazer nada a partir dele mesmo.

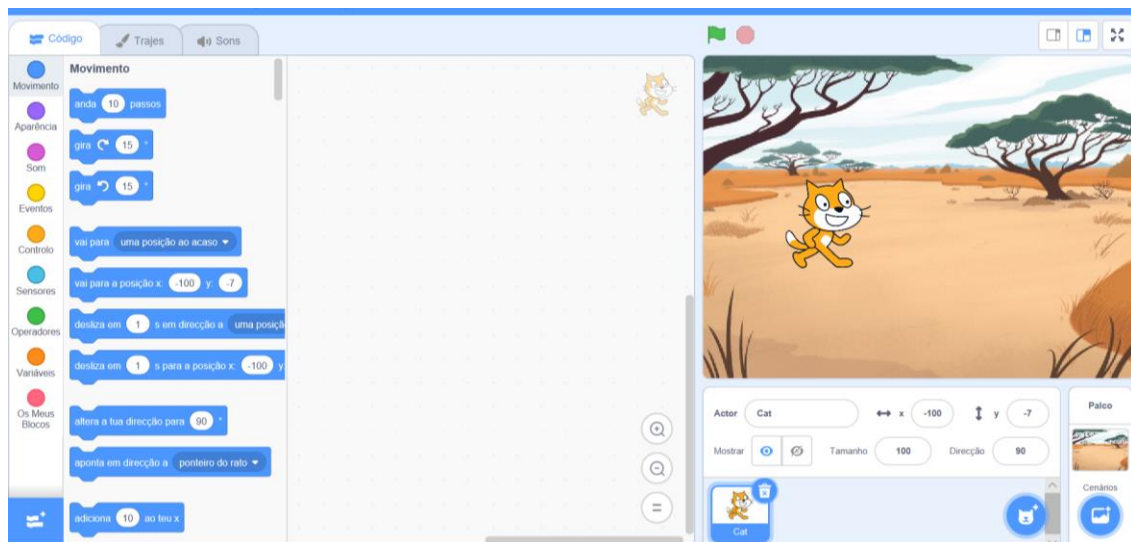
A acção do actor provém duma reacção a guiões introduzidos pelo utilizador na **Zona de guião** (guiões ou códigos de programação). Estes guiões são as instruções ou comandos que indicam ao actor o que ele tem de fazer e elas estão escritas por sequências. O utilizador coloca faixas de código individuais deslizando da paleta de blocos até a zona de guião. Estes blocos encaixam-se logo como pedaços de briquetes dos jogos de construção infantis para criar as instruções.

Vamos entrar então no Scratch e programemos o actor para ele falar e se deslocar no ecrã !

Para abrir o **Scratch** no seu aparelho , dêem dois cliques no ícone Scratch de seu computador.

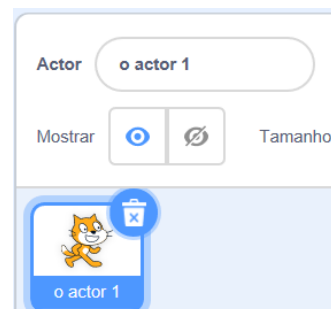
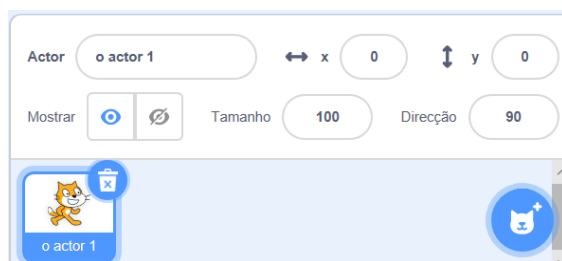
Vão ver o ecrã de começo.





Podem levar o tempo necessário para acostumar-se com as principais funcionalidades deste ecrã ou com o que chamaremos a interface Scratch (homepage).

Observem que a versão reduzida do gato aparece em azul destacado, o que significa que se trata de um componente ativo.



### Instalação do texto

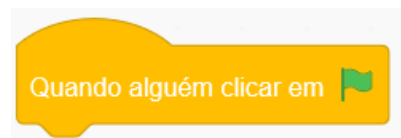
Ao colocar blocos na Zona de Guião, o actor nos dará seu nome.

Vão primeiro na secção **Tipos de blocos** situada na esquerda da interface Scratch, que contém pastas temáticas de blocos de código tais como Movimento, Aparência, Som, etc.

Na direita desta secção da interface utilizador gráfico, está a **Paleta de blocos** que contém todos os blocos de código individuais.

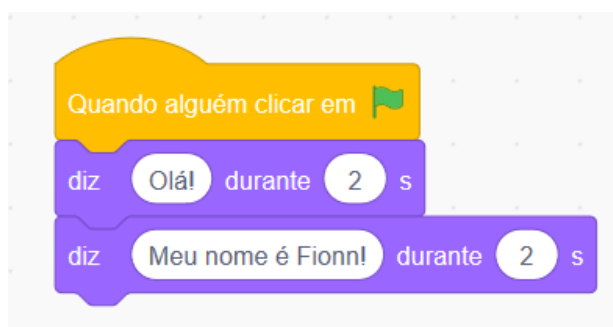


Vão até a pasta **Eventos**.  
Deslizem o bloco seguinte na Zona de Guião :



Este bloco é um comando « Quando ».  
Isso significa que uma vez que foi clicado na **Bandeira verde** que está por cima do Palco, o Actor seguirá os comandos colocados na zona de guião.


Vão até a secção **Aparências** situada no painel Pastas.  
Selecione duas vezes o bloco **dizer Olá durante dois segundos**.  
Digitar o texto Olá! e Meu nome é Fionn( ou o nome que desejar) antes de carregá-los na zona de Guião



Verifiquem que os blocos se encaixam.

Por que não personalizar o actor dando-lhe um nome.

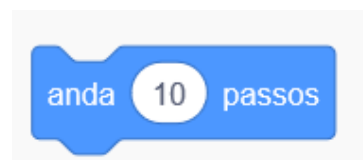


Para fazê-lo, cliquem no ícone  que está na parte superior esquerda da imagem-objecto do gato e digitem o texto Fionn ou outro no quadrado correspondente.

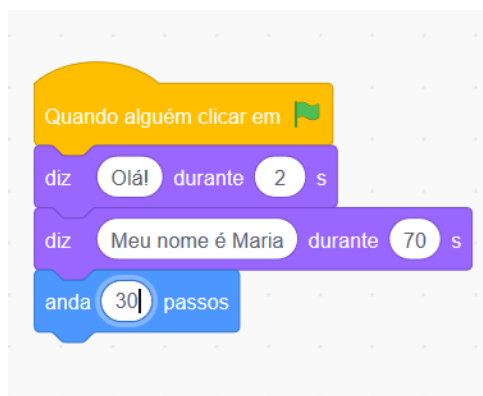
Cliquem na Bandeira verde na parte superior esquerda do ecrã do computador e vejam o que acontece.

Agora, precisamos mover o gato.  
Vão até o arquivo Movimento.

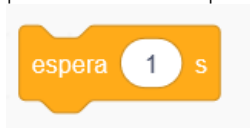
Selecione o bloco Andar e mudem o número de passos chegando até 50.



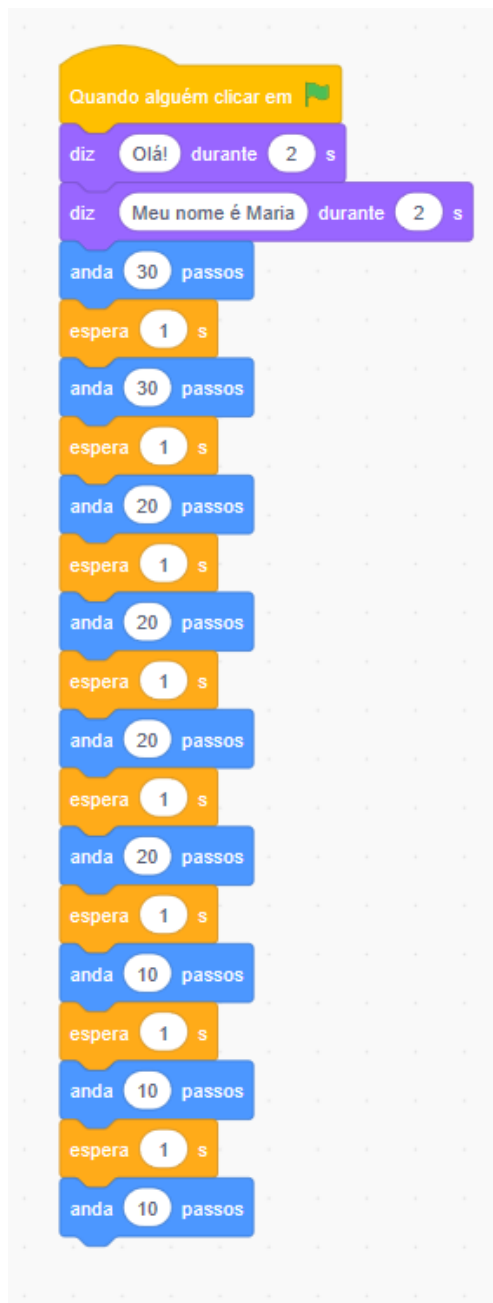
Juntem este bloco com os outros blocos na zona de guião e comecem a programação.



Para aumentar o movimento do actor e para dar-lhe uma impressão de andar, vão primeiro ao arquivo de Controlo e seleccionem o bloco Esperar.



Coloquem este bloco na zona de guião acrescentando alguns blocos Andar:



No entanto, deparamo-nos com uma dificuldade devida ao actor.

Como puderam reparar, se continuarmos com o uso do guião, o gato vai continuar andando até desaparecer quase totalmente do ecrã.

Precisamos portanto inserir um comando adicional com o qual poderá voltar no centro do ecrã cada vez que usarmos este pacote de instruções.

O ecrã está dividido em **coordenadas X** (horizontal) e **Y** (vertical) indicadas no centro do ecrã para que seja **(X)0 (Y)0** e que os números sejam positivos ou negativos conforme seu posicionamento.

Peçam a seus alunos que desloquem o actor no ecrã e que observem como os valores das coordenadas X e Y variam bem acima do ícone do actor situado na parte direita da interface conforme o posicionamento do actor.

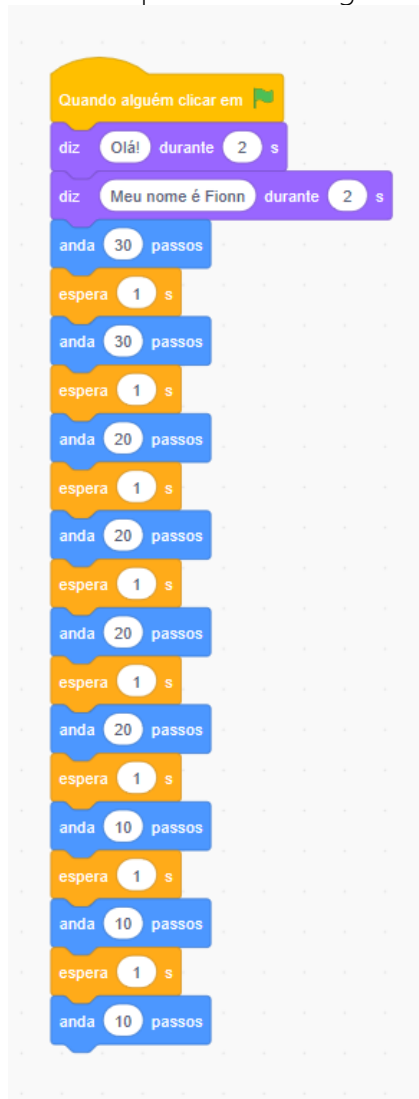


Esta funcionalidade dá a possibilidade ao utilizador de posicionar diferentes actores em diferentes lugares.

(Vejam mais além , o quadrado descritivo sobre a Geometria).

Assim, podemos colocar uma faixa de código no início dos comandos que vai indicar ao Gato que ele volte para o centro do ecrã cada vez que seleccionarmos a bandeira verde.

O texto aparecerá da seguinte maneira :

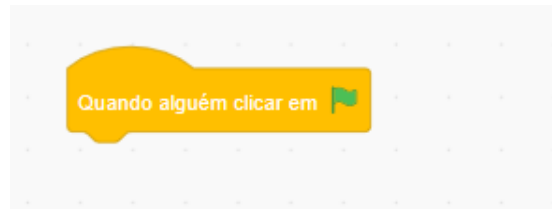


Testem a eficácia desta nova faixa de código ao usar o rato para posicionar o actor para baixo ou para a parte superior do ecrã antes de clicar no ícone da Bandeira verde.

Além da Bandeira verde , outros métodos podem ser usados para começar o guião com o comando « Quando » na pasta de Controlo.

Por exemplo, ao usar a tecla de Espaço ou as teclas de Seta.

Agora, substituem



por



(desde a categoria Eventos ) na Paleta .

Agora cliquem na tecla de Espaço no teclado do computador para iniciar a programação.

Substituem



por



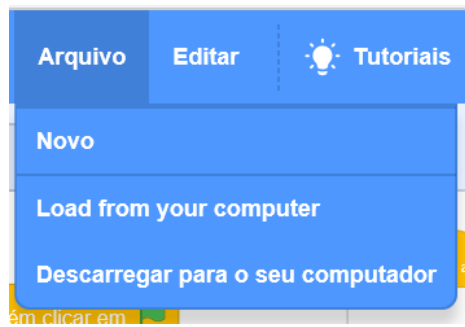
E começar a programação ao clicar no Gato.

Voltem ao bloco Bandeira verde.

## Salvar o projecto

Vão até a pasta Arquivo.

Selecione « Descarregar para o seu computador »



Digite Fionn ou o nome escolhido para seu gato na secção **Salvar como ...**

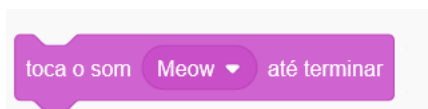
E escolhem o lugar onde querem gravar, tal como aparece no ecrã ou em Meus Projectos .

## Lição 3 - Inserir sons em guião

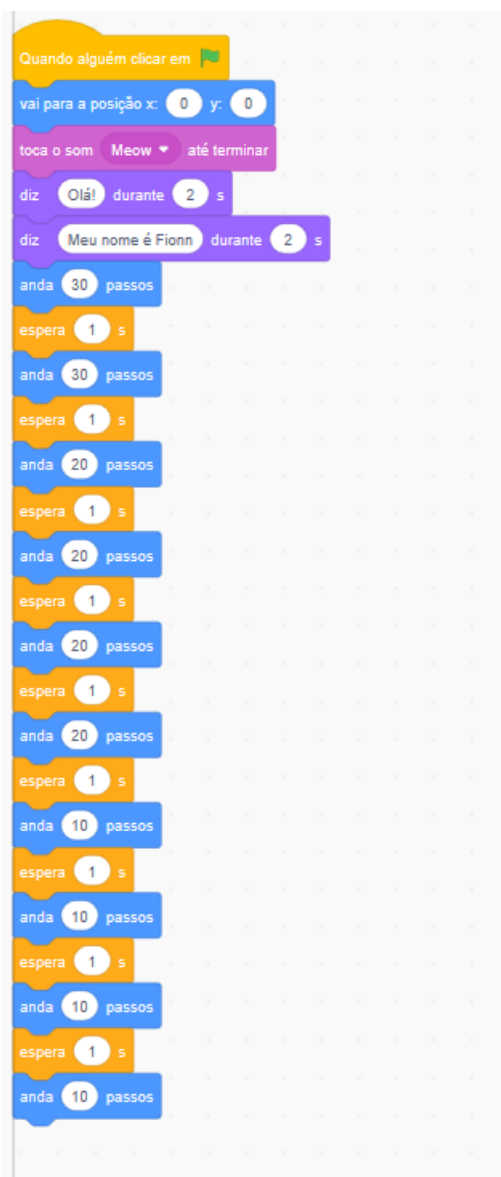
Fazer ronronar o gato !

Vão até a categoria Som.

Selecione o bloco seguinte :



Coloque-o no pacote de instruções na zona de Guião:

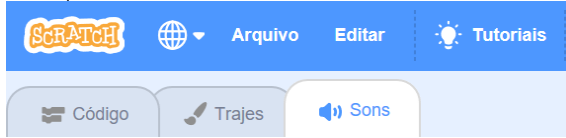


Comecem a programação.

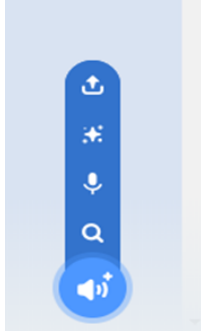
Agora substituem o Meow do gato por outro som de bicho no guião.



Isso pode se realizar ao entrar na livreria dos Sons.



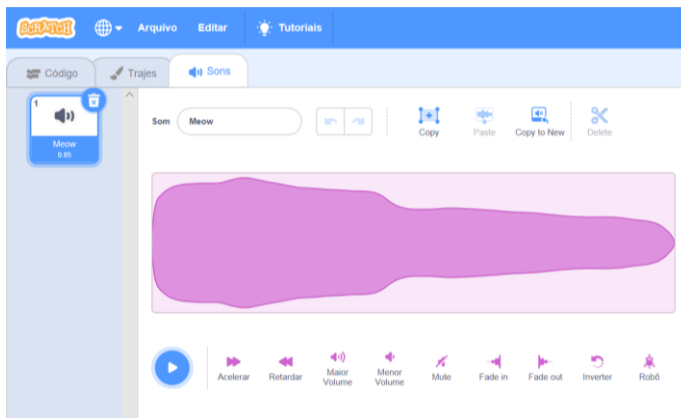
E ao clicar no ícone alta-voz que está na parte inferior da página na esquerda.



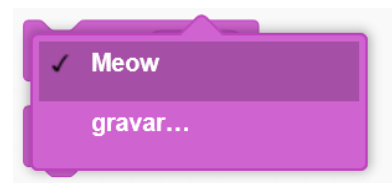
Ali aparece um repertório inteiro de sons.



Cliquem no som escolhido, ele aparecerá logo na lista de sons que fica na coluna esquerda (por baixo do Código cabeçalho) .



Agora podem inserir o novo som no seu guião e escolher o bloco interessado (e a opção de som) gravado como a livreria Código.

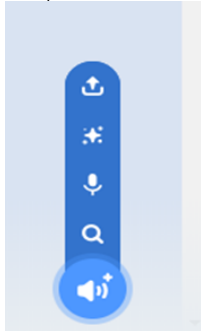


## Exercício

### Waow ! Um gato que fala !

Agora , vamos pedir ao gato que diga as palavras que escrevemos na programação, ou seja Olá e Meu nome é Fionn.

Cliquem no ícone Micro na selecção dos ícones da parte inferior do ecrã na esquerda, como mostra a ilustração seguinte :

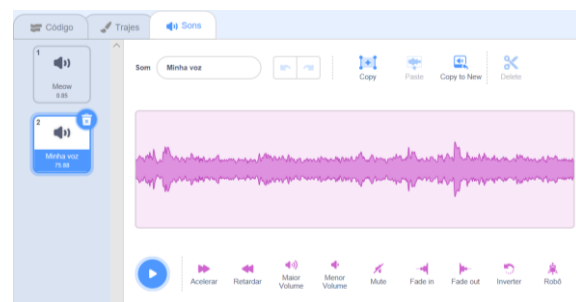


E escolhem a gravação da sua voz ao clicar no botão **VERMELHO** que fica por baixo :



Digam « Bom dia » em voz bem alta. Podem afinar e editem o audio como quiserem .

Passem o título de Gravação 1 em Minha Voz.



Voltem à Zona de Guião e seleccionem a categoria Som( cor de rosa).

Deslizem o bloco Tocar o som da Minha Voz no guião que fica por cima do bloco toca o som "Tropical Birds" durante 2 segundos.

Repitam este procedimento de gravação para Meu Nome é Fionn.



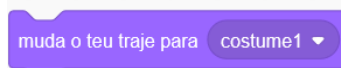
## Lição 4 – Melhorar a marcha do actor

Agora vamos tornar os movimentos do gato mais realistas fazendo suas pernas moverem-se ao caminhar.

Isso pode ser realizado mudando o aspecto físico de um actor individual indo à secção Traje.

Vão para Traje e ali observarão que existe uma segunda imagem ou Traje do mesmo actor com as pernas e os braços em postura distinta que a da primeira imagem.

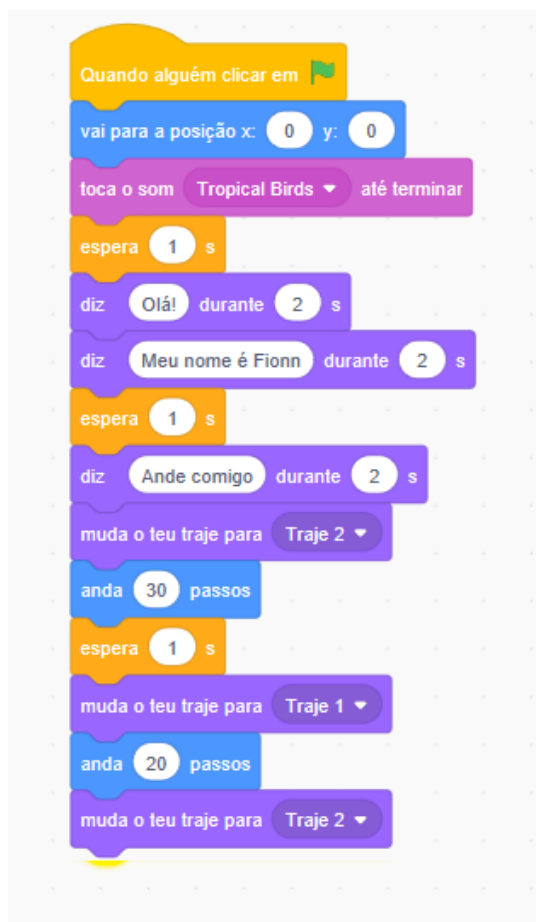
Depois dos primeiros blocos « Andar » e « Esperar » no guião, colocar um bloco « Muda o teu Traje » com Costume2 seleccionado desde a livraria das Aparências.



A opção Traje2 escolhe-se ao clicar na tecla de seta branca à direita deste bloco

E, para os blocos seguintes « Andar » e « Esperar », passe « Alternar no Traje » com Traje1.

Repitam este procedimento ao longo do guião, alternando assim entre Traje 1 e Traje 2.

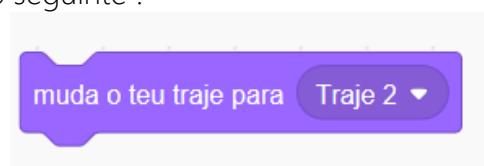


Comecem o Guião ao clicar no ícone Bandeira verde.

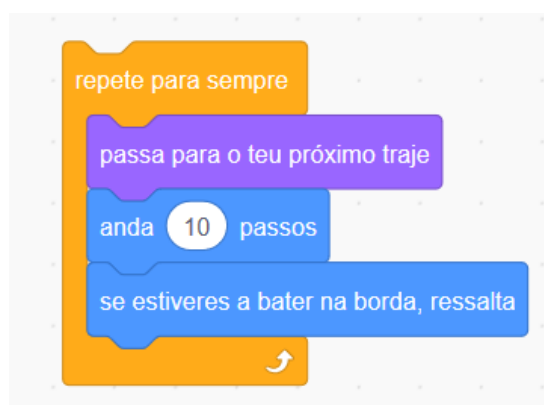
## Um gato muito veloz !

Nesta lição, os estudantes aprendem a usar o bloco muito importante **Repetir para sempre**. Este bloco mostra como o gato pode deslocar-se rapidamente pelo ecrã sem parar !

Primeiro, comecem por separar todo o texto do código colocado por baixo do primeiro evento do bloco seguinte :



No lugar dos blocos que foram removidos, coloquem os novos comandos seguintes :

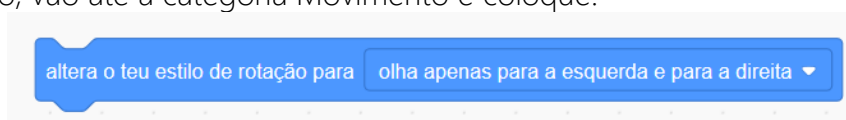


Expliquem aos estudantes a importância do « Repetir para sempre » na programação de computador.

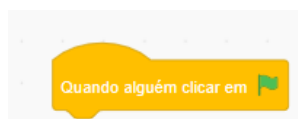
Esta faixa do código indica que o actor trocará de Traje e andará dando uma série de passos, não apenas uma vez mas indefinidamente. Além disso, a adição do comando **se estiveres a bater na borda, ressalta** indica que o actor não desaparecerá do ecrã mas voltar-se-á cada vez que alcançar a borda do ecrã para continuar a andar.

Porém, vão constatar que o actor talvez caminhe agora de cabeça para baixo !

Para ter certeza que o gato fica bem reto e que ele se desloca com a cabeça no lado certo, vão até a categoria Movimento e coloque:



Depois do bloco seguinte :



## Lição 5 - Actores multicoloridos

### Um gato camaleão ! Mudar a cor dum actor

Mudemos a cor do gato enquanto ele caminha .

Para realizá-lo , vão na opção Traje, seleccionem o ícone do gato Traje1 logo com o clicar direito do rato seleccionem Duplicar.

Façam a mesma coisa com o ícone Traje2.

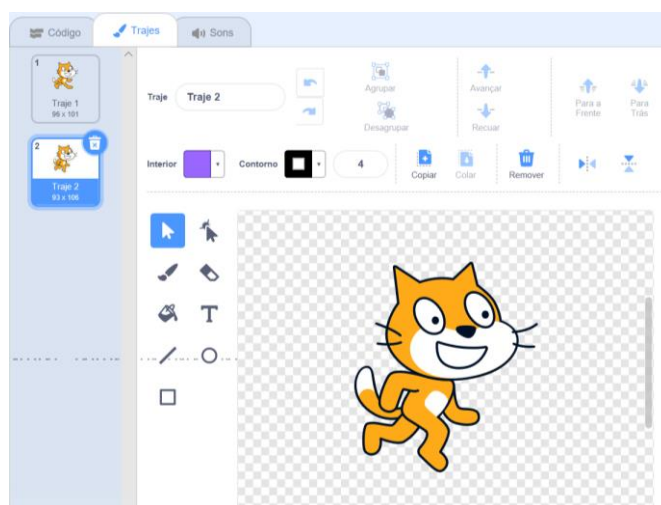
Repitam este procedimento para cada Traje 1 e 2 até obter nove Trajes.

Modifiquem os números de Traje no guião para eles aparecerem por ordem cronológica .

Logo cliquem no ícone Traje 2



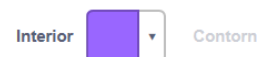
No lado direito dos ícones Traje encontra-se o **Editor de Desenho**, que permite colorir o actor existindo e também desenhar e colorir novos.



Escolhem uma cor diferente na paleta chamada **Interior** situada na parte superior na esquerda do ecrã .



Escolhem uma cor e usem o ícone lata de tinta para incluir esta nova característica no Actor existindo no ecrã .



Certifiquem-se que o Editor de desenhos esteja no **Modo Bitmap** ao usar a ferramenta lata de tinta.



Observação :Apresentem aos seus estudantes algumas das demais funcionalidades do Editor de Desenho,especialmente as ferramentas de duplicação, de texto e de supressão.

Quando terminarem pressionem no ok.

Repitam este procedimento para incluir uma nova cor em cada Traje que fica .

Comecem o Guião com um clique no ícone Bandeira verde e observem o andar do gato de múltiplas cores !

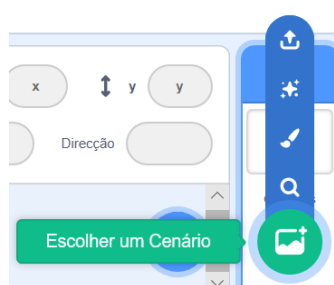
## Lição 6 - Modificar o palco

Pelo momento, estamos a usar um cenário branco para o gato .  
Vamos trazer algo mais interessante na vida do gato com uma animação dentro de uma nova paisagem.

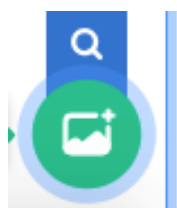
Ao observar a parte inferior do ecrã na esquerda , reparamos que o gato Fionn aparece em destaque azul, o que significa, como foi indicado anteriormente que o Actor é o elemento de acção no Guião.

Cliquem em Palco.

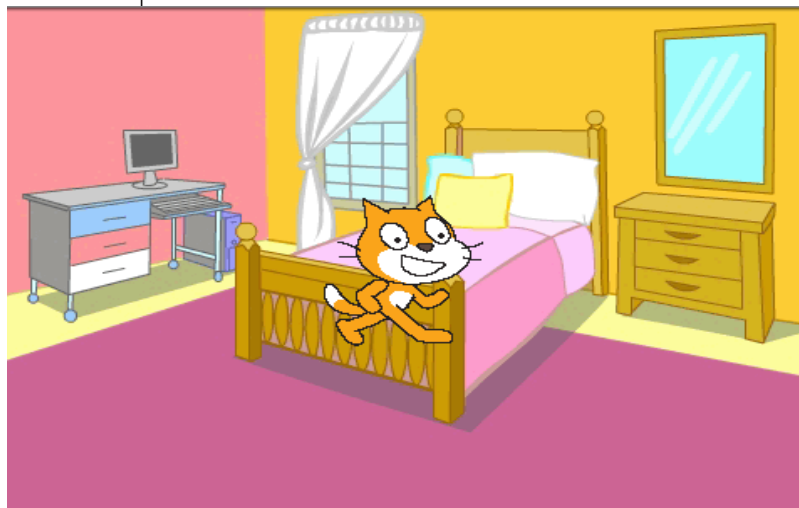
O ícone Palco fica agora em destaque azul o que indica que se tornou o elemento ativo que agora pode ser modificado pelo utilizador.



Vão até o ícone Escolher um cenário situado por baixo do ícone Palco e seleccionem uma das cinco opções na linha dos ícones (mostrado abaixo).



Vão até a pasta chamada Interior e seleccionem Bedroom 2.



Clicquem no ícone Bandeira verde para iniciar o programação.

Vão observar que surge um problema, parece que o gato Fionn está caminhando pelo ar !

Peçam primeiro aos alunos proporem uma forma de resolver a maneira de codificar o guião para fazer com que o gato seja movido sobre o chão do quarto.

Claro, teremos que adicionar alguns códigos de comando no guião do gato para ter certeza que ele está caminhando no chão.

Então dar dois cliques no ícone Cenário para voltar ao Guião do actor (gato).

Desloquem o Actor para o canto inferior da parte esquerda do ecrã Palco .



## GEOMETRIA

As imagens de computador são compostas de pixels que podem ser identificados por coordenadas X e Y.

Um ecrã de computador ou uma imagem é constituída de unidades de base chamadas **Pixels**. É Uma abreviacção de duas palavras , imagem e elemento.

Os pixels são as menores unidades de cor num ecrã ou numa imagem de computador e que podem ser controladas ou programacçãoadas.

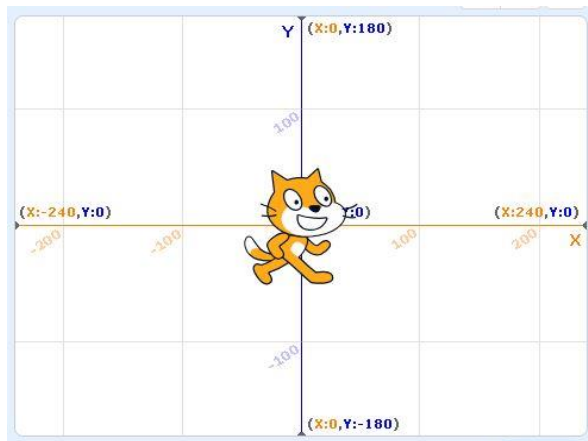
As coordenadas X, Y são respectivamente os endereços **horizontal(X)** e **vertical (Y)** de todos os pixels ou ponto de localizacção numa visualizacção de ecrã de computador.



A coordenada  $x$  é um número dado de pixels ao longo do eixo horizontal de uma visualização no ecrã a partir do pixel ( pixel 0) do ecrã central. As coordenadas  $x$  e  $y$  localizam conjuntamente um lugar de pixel específico no ecrã. As coordenadas  $X$  e  $Y$  fazem parte do ramo das **matemáticas** conhecida pelo nome de **geometria** que aborda os assuntos de posição relativa das figuras, das formas, dos tamanhos e das propriedades do espaço.

Para acostumar os utilizadores com as grelhas  $XY$ , acessem a nova secção Cenário, como feito anteriormente, seleccionem o último ícone ou ante-penúltimo ícone na fila dos cinco ícones. E escolhem a grelha  $XY$ , ante-penúltima opção de ecrã da biblioteca de cenário.

Aparece a seguinte visualização de ecrã :



Permitam que os estudantes vejam o ecrã e que eles possam deslocar o actor com o fim de observar as mudanças de coordenadas.

Expliquem sua importância, especialmente nos jogos de computador (por exemplo deslocar objectos num jogo até uma posição de partida).

Voltem ao palco do quarto 2 .

Desloquem o gato no canto superior direito do palco. As coordenadas  $X$  e  $Y$  do gato aparecerão debaixo da cena da forma seguinte :



Uma vez que o gato estiver posicionado , as coordenadas  $X$  e  $Y$  atuais do gato aparecerão nos quadrados  $X$  e  $Y$  ( como na imagem ilustrada abaixo )  
Vão até Movimento.

Visualizem o bloco **Ir a X: Y:**

Deslizem este bloco na zona de guião e coloquem-no diretamente por baixo do primeiro bloco( Bandeira verde) do pacote das instruções da programação. Como mencionado abaixo, as coordenadas  $X$  e  $Y$  do gato serão indicadas no bloco.

Cliquem no ícone Bandeira verde para iniciar a programação .

Pergunta : o que devemos fazer para que o gato pule na cama ?

É claro que o operador precisa pedir que o actor vá para a cama e que coloque no lugar certo da programação , um bloco de movimento incluindo as coordenadas X e Y certas.

Primeiro, comecem a programação.

Vejam o lugar onde o gato pára e a programação termina.

Desloquem o gato na cama.

Vão até Movimento.

Visualizem o **Desliza em 1 s para a posição X: Y:** bloco

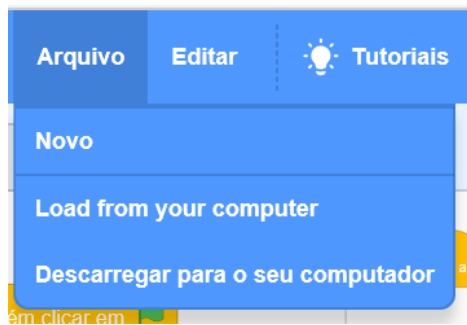
As novas coordenadas X e Y (do gato na cama ) serão visualizadas no bloco de código. Deslizem este bloco na Zona de guião e deixem-no na localização certa no conjunto de instruções do programação (será melhor inseri-lo no fim da programação) .



Cliquem no ícone Bandeira verde para começar a programação.



Salvem seu arquivo de programação com o nome GatoPula seleccionando **Descarregar para o seu computador** no menu abrindo **Arquivo** situado na parte superior do ecrã Scratch.



### Exercício

Deslizem alguns blocos adicionais de código na programação para permitir ao gato que pule na cama e que ande alguns passos.

Além disso, certifiquem-se que o gato mude de cor cada vez que der um passo .

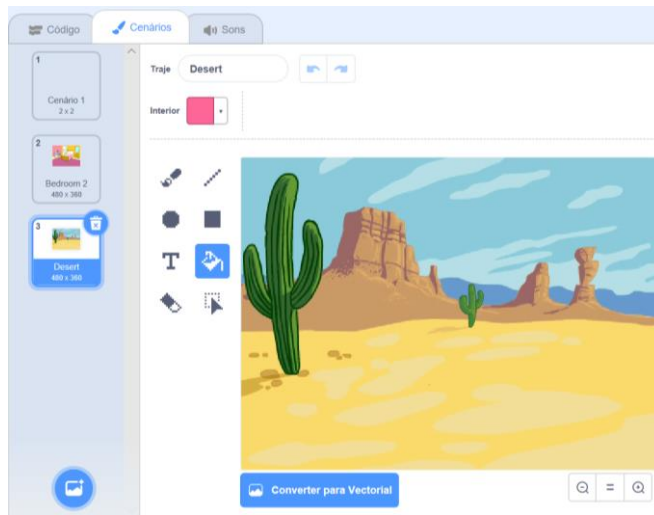
### Exercício

Substituem o cenário por uma nova imagem da biblioteca de cenários.

Reparem que novo cenário aparece numa lista de cenários do lado esquerdo do ecrã para realizar seu projecto.

Para voltar a um cenário anterior ou a um novo, basta que coloquem o cursor sobre a imagem de cenário apropriada e cliquem.

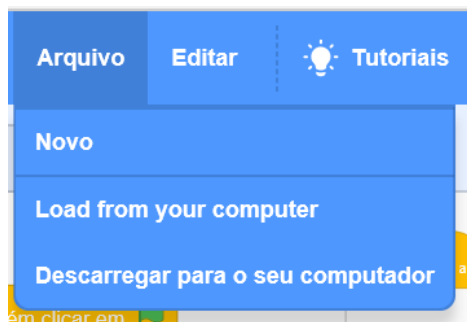
A nova imagem fica agora em destaque, é o elemento ativo e ele torna-se o cenário que aparece no ecrã Cenário.



Logo repitam as instruções para que o actor pule sobre e desde um objecto que aparece na imagem seleccionada.

Observação : o utilizador deve clicar no actor do gato antes que fique em destaque (azul) quando o código pode ser adicionado.

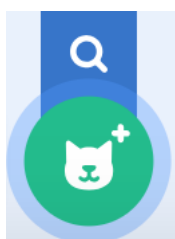
Salvem seu novo arquivo programação com o nome GatoPula2 ao seleccionar **Descarregar para o seu computador** no menu abrindo **Arquivo** situado na parte superior esquerda do ecrã Scratch.

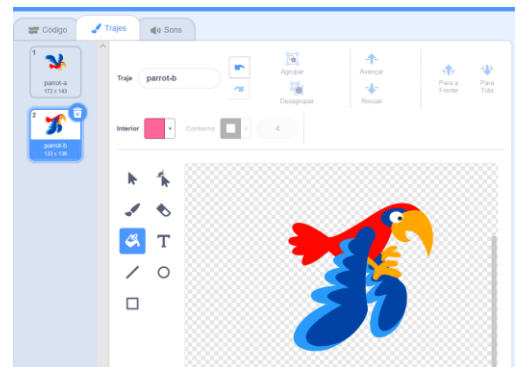


### Acrescentando uma ave que voa

Esta funcionalidade realiza-se colocando uma ave a voar continuamente pela sala, perto do teto, para que fique inacessível pelo gato.

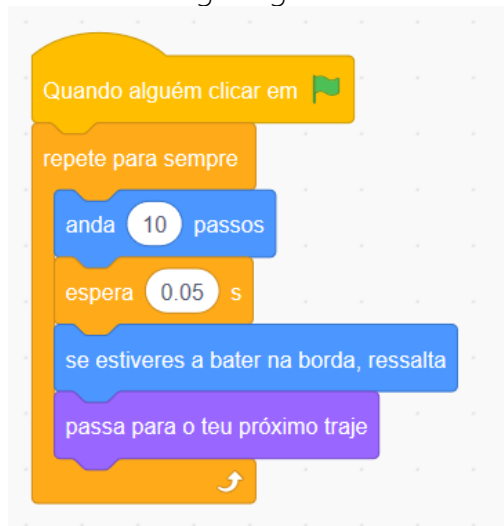
Selecione um novo actor clicando em qualquer ícone ilustrado abaixo :



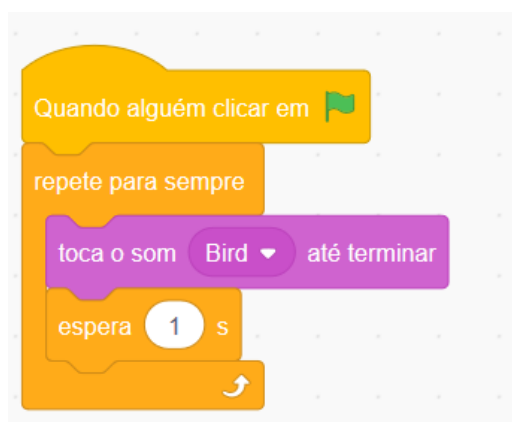


Selecione uma ave com duas trocas de Traje (para dar uma impressão de movimento de asa).

Insiram o código seguinte :



Para adicionar o som , insiram o guião adicional seguinte :



## Exercício

Caminhar numa estrada.

Desenhem um novo actor , por exemplo um menino ou uma menina.

Desenhem diferentes Trajes (versões).

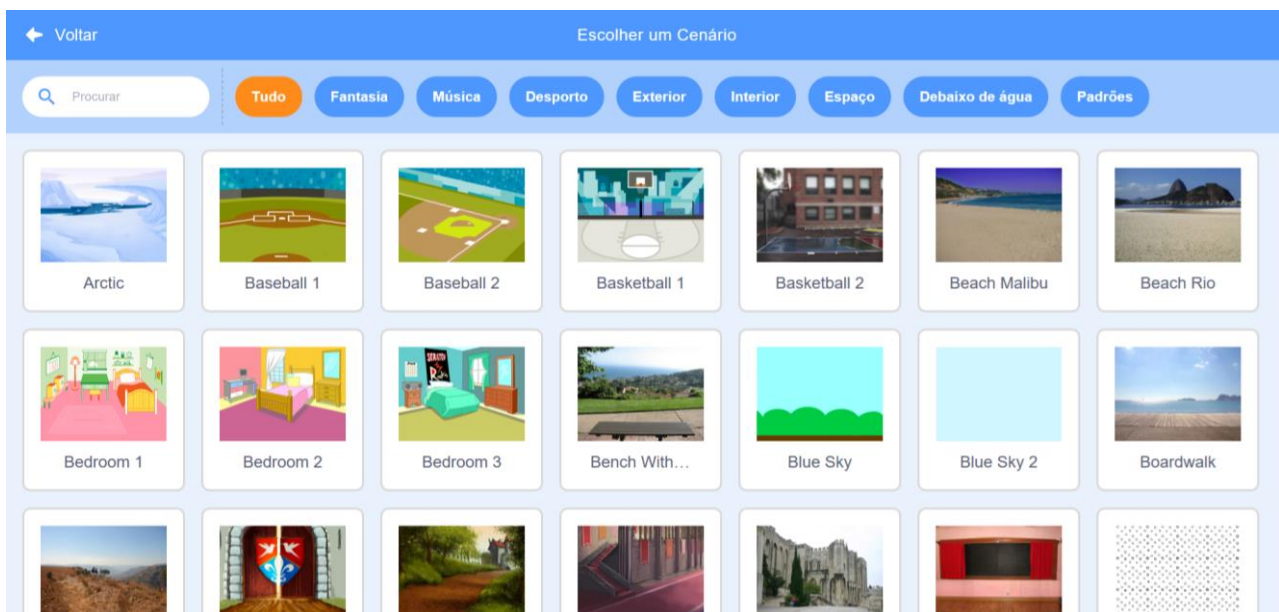
Desenhem uma cena de rua (cenário).

Escrevam um programação para ter um menino ou uma menina caminhando numa rua.

Desenhem outros actores com guiões também para caminhar, correr e fazer bicicleta.

Insiram uma ave que voa constantemente em movimento de vai e vem no céu.

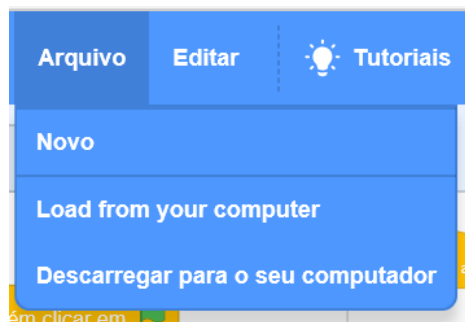
Observação : antes de abordar o exercício abaixo , mostrem aos alunos como pode importar-se uma foto como « cenário » a partir da internet ou a partir do computador, e como ela pode ser modificada ao usar as ferramentas do **editor de desenho**.



## Lição 7 – Um cão que late e persegue o gato !

Programemos um cão que late ao caçar Fionn, nosso gato querido !

Vão até o menu Arquivo



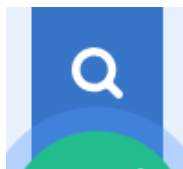
E escolhem **Load from your computer**

Logo cliquem em **GatoPula2** para abrir seu arquivo Scratch anterior

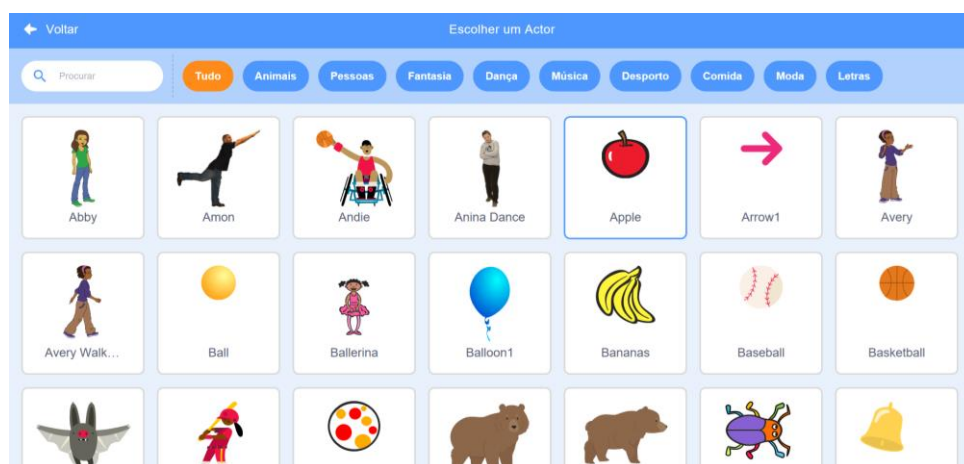
Vão ao



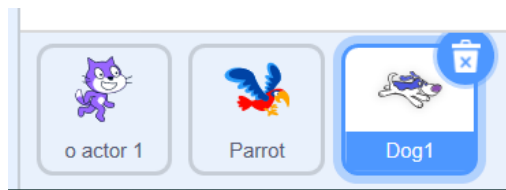
Seguido de



O ecrã seguinte aparece:



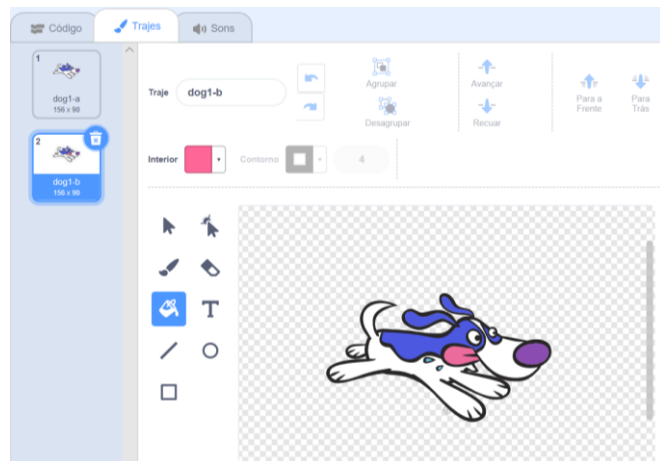
E vão até a pasta Animais e escolhem a imagem do primeiro cão que corre (Dog1). Observem que o ícone do Cão está agora em destaque azul no contorno indicando que é o actor que está ativo.



Vamos construir agora um guião ou um programação para animar o cão que possa correr e falar.

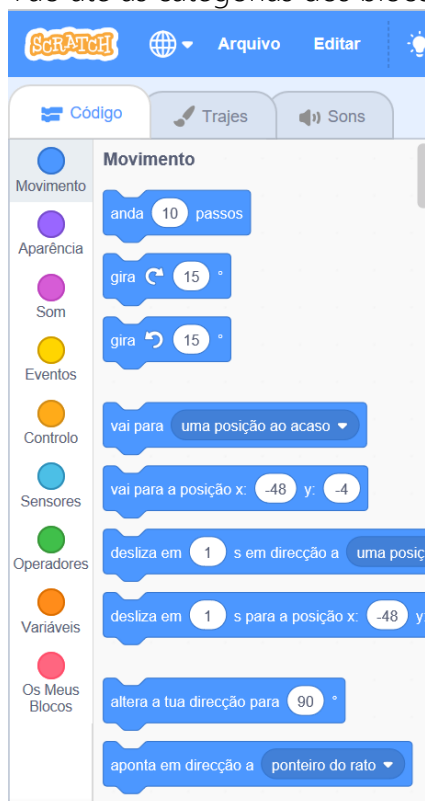
Para isso, precisamos ao menos de outra versão do cão que, com o actor de cão atual, dará uma impressão de movimento.

Ao seleccionar o Traje, vê-se que já existem duas versões possíveis de serem usadas no programação.



### Construir um guião para animar o Cão

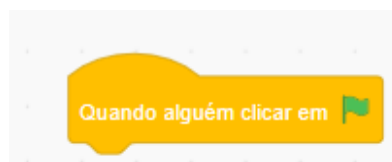
Vão até as categorias dos blocos que estão na esquerda do ecrã Scratch.





Cliquem na pasta Eventos

Insiram o bloco seguinte na zona de guião :



Isso indicará que quando clicar-se-á na Bandeira verde por cima, o actor seguirá os comandos inseridos na zona de guião.

No entanto, lembrem-se que agora estamos com dois actores , ou seja o gato e o cão.

Queremos que o cão apareça no palco (ecrã )apenas depois que o gato tenha pulado na cama.

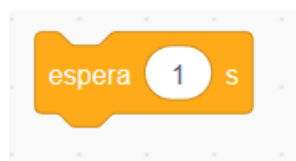
Portanto, precisamos escondê-lo quando o programação começar e fazer com que ele apareça (ser visível) na cena apenas em determinado local ( Xe Y) e apenas depois de determinado momento (esperar).

O operador poderá efetuar esta tarefa com os blocos seguintes, em sequência de combinações :

Os blocos Mostra-te e Esconde-te na categoria Aparência:

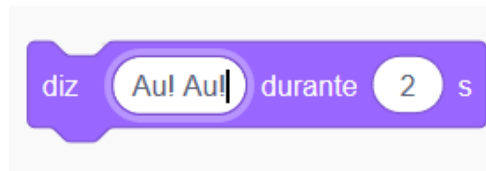


E o bloco Espera na categoria Controlo

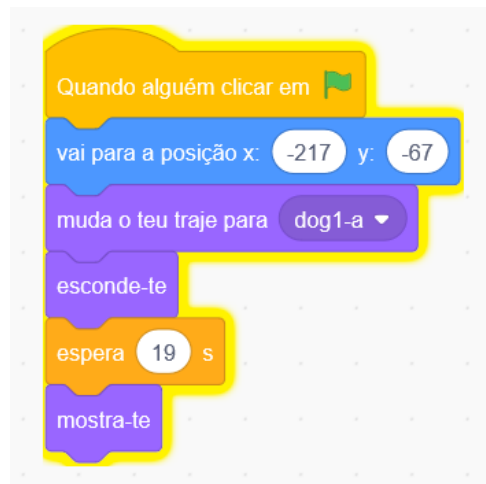


Assim como o bloco das coordenadas X e Y na categoria Movimento. As coordenadas devem corresponder com uma posição na extremidade esquerda do gato (no mesmo eixo X) e perto da borda do ecrã.

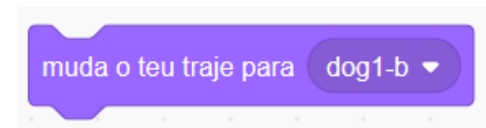
Mande o cão latir seleccionando o bloco Dizer\_\_\_\_\_ na categoria Aparências.



Para dar um movimento ao actor, precisamos primeiro inserir um bloco « Alternar para o Traje » no início do guião para que a primeira versão do cão, ou seja dog1-a, apareça sistematicamente quando o programação começar.



Depois deste primeiro Movimento ( neste caso o bloco Ir a x\_\_\_ e y\_\_\_) e o bloco Esperar , situado no guião, deslizem um bloco « Alternar para o Traje » com dog1-b tomado na categoria Aparências.



A opção dog1-b escolhe-se clicando no ícone do triângulo branco situado na parte direita deste bloco.

E para os próximos blocos Andar e Esperar, usem um « Alternar para o Traje » para ativar o bloco Traje com o dog1-a.

Repitam este procedimento durante todo o guião , alternando consecutivamente entre dog1-a e dog1-b.

O resultado deveria aparecer como na ilustração abaixo :



### Um gato assustado !

Para dar ainda mais autenticidade ao projecto, o gato Fionn deve gritar « Socorro ! » e mostrar um rosto triste quando o cão aparece e antes de pular sobre a cama.



Enquanto ele está na cama, o rosto do gato poderia ficar cheio de lágrimas !

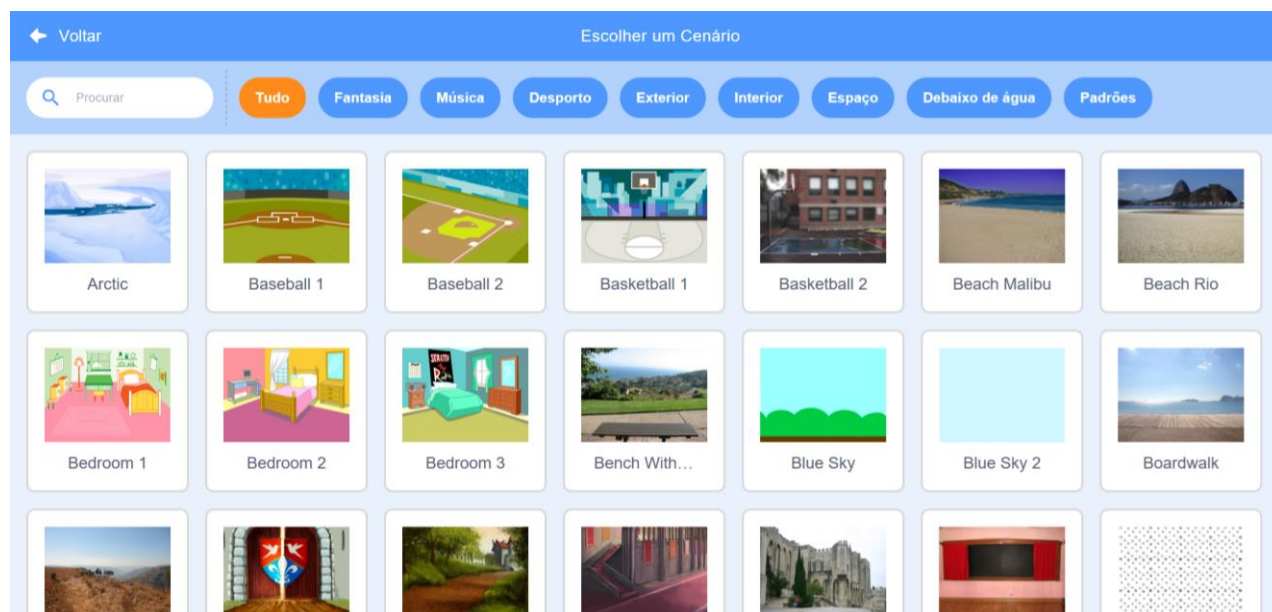
Portanto, peçam a seus alunos que criem uma mudança de Traje adicional para o gato e que insiram o código adequado na Zona de Guião.

## Lição 8 – Criar um actor

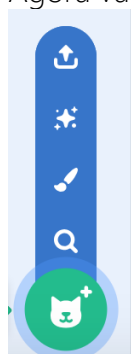
Nesta lição, os alunos vão acostumar-se com o editor de desenho potente de Scratch, que permite aos utilizadores criarem seus próprios actores.

Cliquem no ícone Cenário na secção da parte inferior direita do ecrã e seleccionem a cena Beach Malibu da secção Fora da biblioteca de cenário.

Podem inserir o actor gato na cena da praia. Mas deslizem-no no canto inferior esquerda do ecrã, pegando um guião que o fará desfilarm na praia.



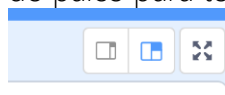
Agora vão até a secção dos novos actores situada na direita por baixo do palco.



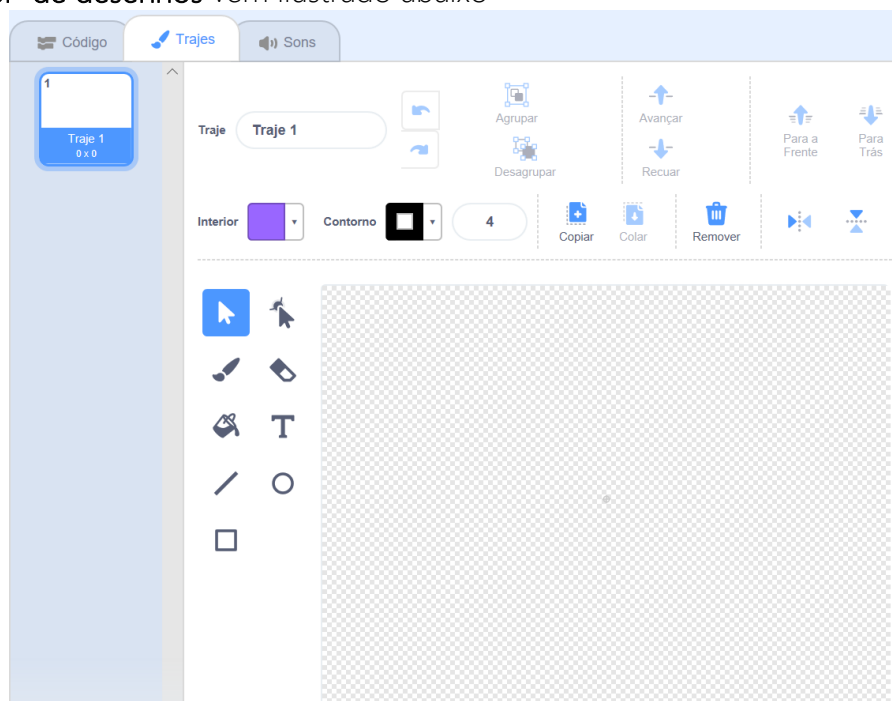
Seleccionem o ícone pincel .



Diminuem o tamanho do Palco clicando no ícone do centro na parte superior direita do palco para tornar o Editor de desenho prioritário.



O Editor de desenhos vem ilustrado abaixo



Fiquem certos que o Editor está em modo Bitmap (na parte inferior esquerda).

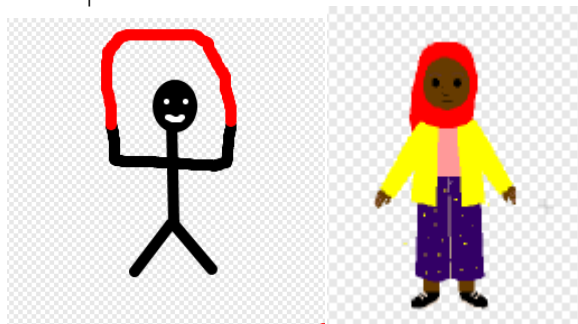
Esta programação de desenho integrada é um sistema de desenho de muitas capacidades e fácil de usar.

Acostumem-se com todas as suas funcionalidades-chaves: paleta de cores, supressão, pincel, etc.

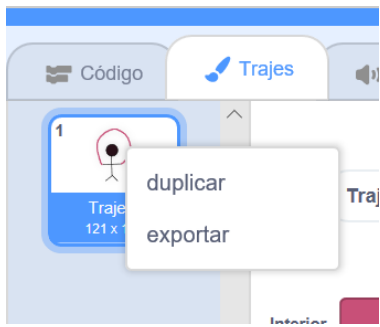
Agora desenhem um actor homem fósforo que pula com o uso da paleta de cores (preencher), opções de círculos, de traço e de pincel. Certifiquem-se que o desenho fica no centro do ecrã cheio de pixels.



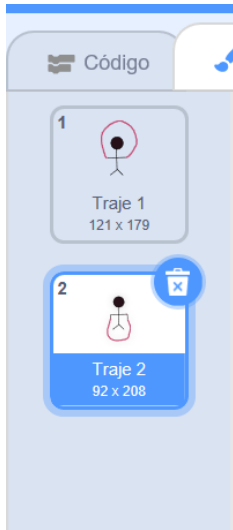
O actor pode ser tão simples ou pormenorizado quanto quiserem. Desta forma, vocês podem acrescentar elementos tais como cabelos, olhos, uma boca, etc.



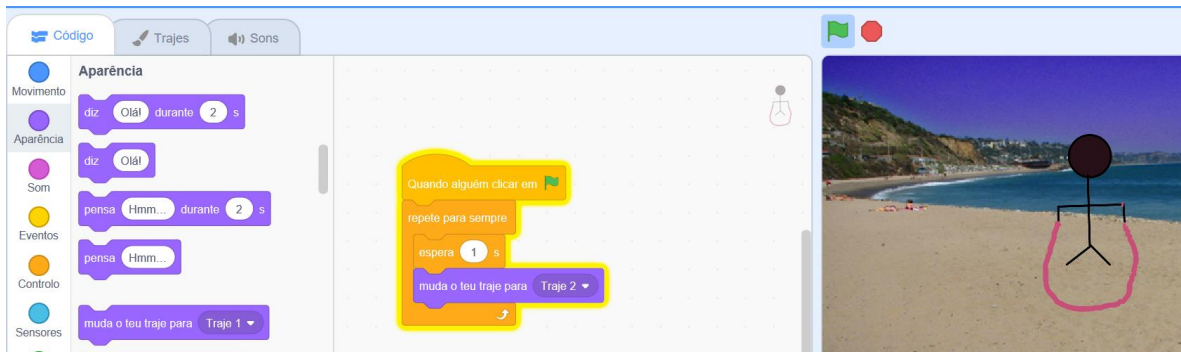
Uma vez terminado a figura-fósforo, vão até a sua pequena representação icónica de Traje. Dupliquem-no ao clicar no teclado ou com um clique direito do rato.



Modifiquem esta segunda versão do Traje até que os braços, a corda e as outras características do actor sejam posicionados de forma distinta da primeira versão.



Quando terminarem, usem um guião simples para animar o actor pulador.

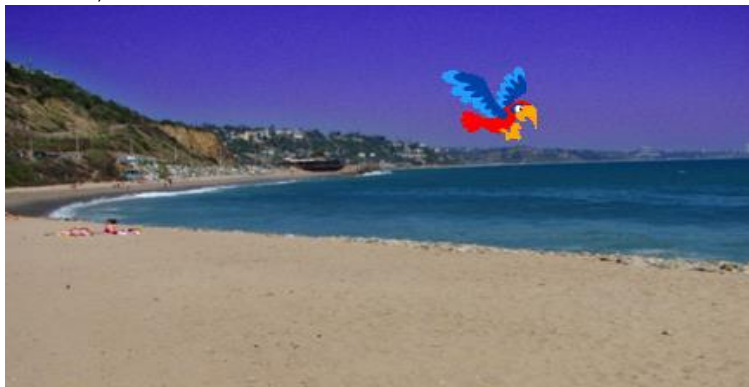


## Lição 9 – Os actores controlados por cursor

Nesta lição, os alunos vão acostumar-se com a programação dos actores para acompanhar os movimentos do cursor.

### Ave tropical na praia

Seleccionem o palco Beach Malibu e um actor com dois Trajes a partir das livrarias cenário/ actor



Use a opção **Ponteiro de rato** situado no bloco **orientar-se para** (Movimento) e construam o guião seguinte :

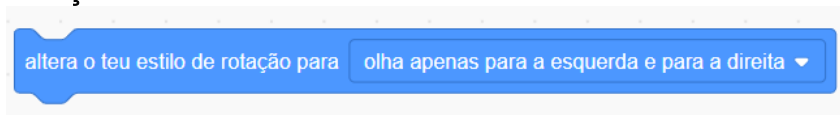


Modifiquem os números no bloco **esperar** \_\_\_\_\_ segundos.

Observem o efeito no movimento do papagaio .

Para impedir que a ave voe em sentido contrário, usem a opção do centro na secção

**Rotação** :



**Exercício** :

1. Desenhem seu próprio actor ave com duas modificações de Traje usando um palco na secção Fora da livraria Cenário.
2. Criar um programação com o actor controlado pelo código Ponteiro de rato.

## Lição 10 – O actor psicodélico

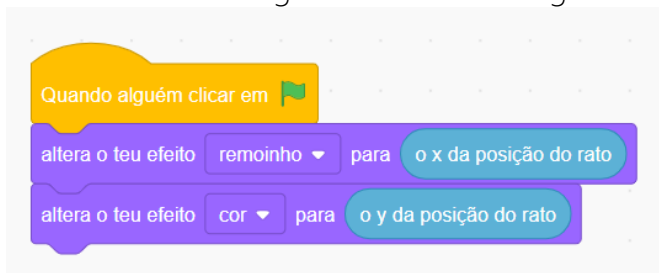
Esta lição apresenta aos alunos a categoria *Sensores* que possibilitará neste caso modificar a aparência física do Actor pelos movimentos do rato ( ou do teclado).

Seleccionem um Actor, com preferência para um corpo multicolorido tal como o Butterfly 2.

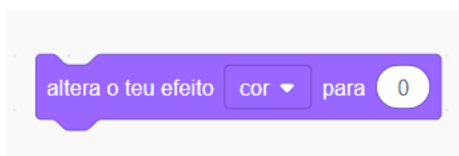


Vão até a secção Código do actor

Insiram na Zona de guião os blocos código como está ilustrado abaixo :



O bloco adjacente, tal como aparece ilustrado no guião abaixo, está situado na categoria Aparências.



Vão até a categoria Sensores para visualizar os blocos Rato X e Y que vão deslizar no bloco **Altera o teu efeito \_\_\_\_\_ para \_\_\_\_\_** como vem ilustrado no guião abaixo.

Comecem a programação com um clique na Bandeira verde na parte superior direita do ecrã.

Pelas instruções enviadas que deslocam o ponteiro do rato no eixo X , o actor mudará de forma. Se deslocarem o rato no eixo Y, o actor mudará de cor.

### Exercício

Testem outros efeitos com um clique na tecla de seta branca no « aplicar o efeito a ...» e escolhem outras opções além de Bobinar e Cores.



## Lição 11 - Interacção do actor

Esta lição baseia-se nos seus conhecimentos de comandos de **Sensores** adquirados na lição 9 para mostrar como codificar uma interacção automática entre dois actores com o bloco de código **Tecla**.

Primeiro apaguem o gato .

Escolhem Max a partir da categoria Pessoas da livraria dos actores.



Logo escolhem o trampolim como segundo actor.

Por baixo da secção Traje de Max, apaguem todas as versões do actor menos max-b e max-c.

Apenas com estas duas versões, será suficiente para animar Max a pular de alto para baixo no trampolim .

Ainda assim , podem usar o Editor de desenho para mudar um ou vários Trajes para visualizar os braços ainda mais altos ou mais baixos. Isso pode ser realizado com o uso da ferramenta em destaque para a ilustração abaixo.

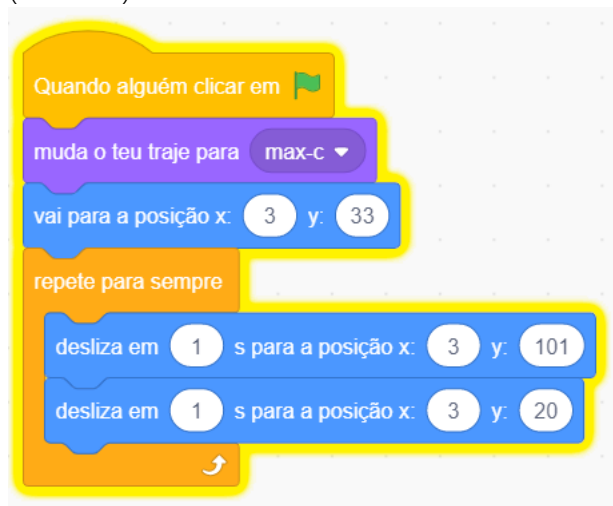


Claro que queremos que Max pule sobre e por baixo do trampolim.

Como foi feito numa lição anterior , damos a impressão de pular usando o bloco deslizado desde a categoria **Movimento**.

Mas desta vez, usamos dois blocos Deslizar pois queremos que ela suba até um determinado ponto (eixo Y) antes de ela descer para tocar o trampolim.

Esta acção realiza-se pelo segundo bloco Deslizar com uma coordenada Y inferior (número).



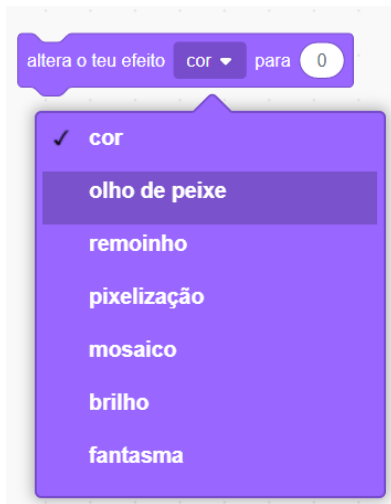
Quando Max tocar o trampolim, seus braços erguerão-se. No mesmo tempo, a área do trampolim irá reagir conforme o peso do Max ao estender-se até que ela pule para cima de novo.

Para realizar a aplicação deste efeito inserimos o comando **Tecla** a partir da categoria **Sensores** nos guiões de Max e do trampolim.

Para Max :



Para o trampolim, o uso do efeito olho de peixe da categoria **Aparências** dá a impressão de um abaulamento.



O guião aparece desta forma abaixo :



Anotem que o efeito olho de peixe deve ser voltado para o zero , no caso contrário o trampolim continuará abaulado cada vez que Max o atingir.

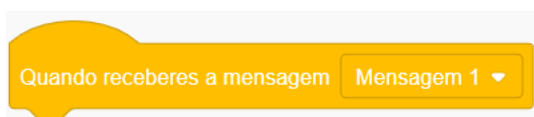
### Exercício

1. Criar um palco completo de área de jogos com um balanço clássico e outro com sistema de inclinação.
  2. Desenhem duas crianças colocadas em cada ponto extremo do balanço com sistema de inclinação.
  3. Dêem uma impressão de movimento ao balanço com sistema de inclinação.
  4. Desenhem uma criança no balanço clássico.
  5. Dêem uma impressão de movimento ao balanço.
  6. Desenhem um menino ou uma menina pulando.
- Estes personagens podem ser simples figuras-fósforo

## Lição 12 - Dois actores estão a conversar

Nesta lição, os alunos vão acostumar-se com o código de comando **Enviar** que manda uma mensagem para outra parte da programação (por exemplo outro actor ou cenário) pedindo que aplique uma modificação.

« Enviar » é usado para facilitar a comunicação entre actores e outros elementos de Scratch. A resposta do actor destinatário será accionada por uma inserção numa faixa de código conhecido pelo nome de



Seleccionem duas vezes o actor Butterfly 2 a partir da livraria de Scratch. Escolhem um cenário adequado.



Observem que se seu actor estiver orientado no lado errado será preciso clicar na aba Traje.

No Editor de desenho, cliquem no ícone Flip horizontal (esquerda-direita) em modo bitmap para inverter a direcção do actor.



Insiram o código seguinte no primeiro actor Butterfly 2



Insiram o código seguinte no segundo actor Butterfly 2



Continuem este tipo de codificação até o fim da conversa.

### Exercício

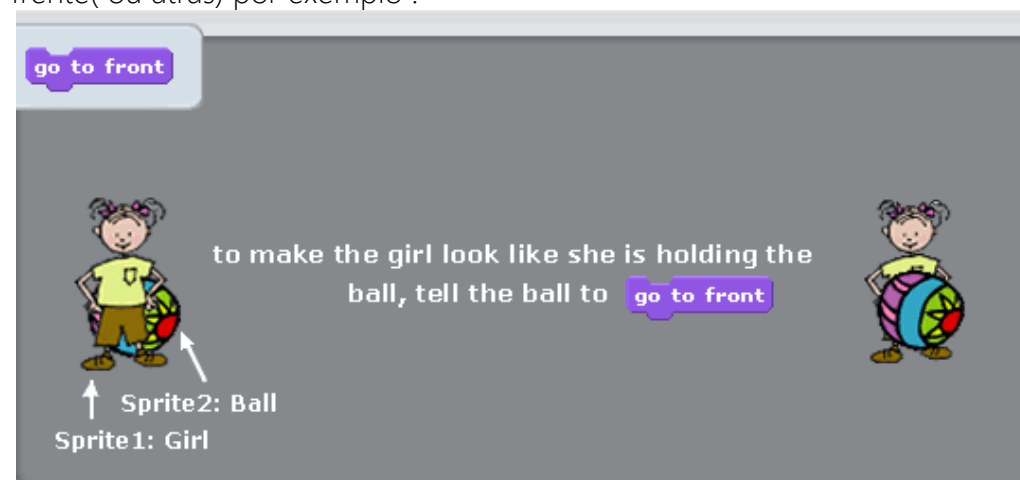
1. Criem uma programação onde duas pessoas ao caminhar estão a conversar depois de uma reunião.

## Lição 13 – Criar um recife de coral

Objectivos da lição :

- Entender o significado do comando **Se** (com seu impacto de causa e efeito) encontrado na categoria **Controlo**.
- Usar o bloco de código **Número Aleatório** da categoria **Operadores** para mudar a posição do actor
- Usar o comando **Sensores** para produzir diferentes efeitos quando os actores entrarem em contato.
- Adicionar uma partição de **música**.

Expliquem que o 'ir para o primeiro plano' é muito útil quando são dois actores colocados um por cima do outro e que vocês querem que um apareça sempre na frente( ou atrás) por exemplo .



Os blocos de código 'ir ao cenário' podem ser aplicados quando vocês têm vários objectos no ecrã tal como um aquário, cheio de vários peixes e plantas e onde têm de representar alguns objectos no fundo enquanto que outros ficariam na frente quando se cruzam.

### 2ª Parte : Criar um mundo marinho

Criemos agora nosso próprio mundo marinho tropical colorido.

No menu Arquivo ir até Novo

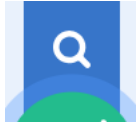
Apaguem o actor Gato

Selecionem o palco Underwater na pasta Mundo aquático



Vão até o menu dos actores

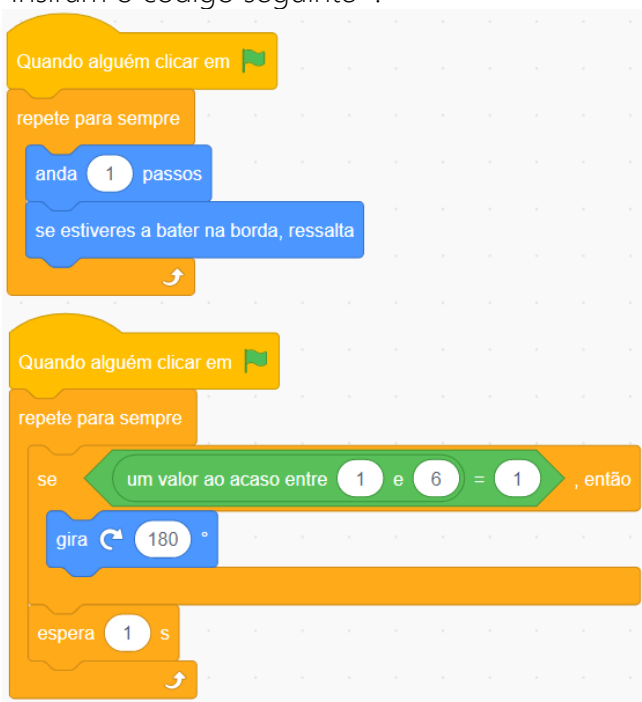
e cliquem em



Para acessar a livreria dos actores.  
Escolhem o actor Shark\_ na pasta Animais.

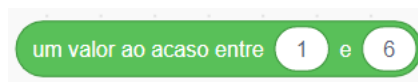
Escolhem o actor Fish na pasta Animais e dupliquem-no.  
Apaguem todos os Trajes menos um para cada peixe. Certifiquem-se de que cada um dos dois actores de peixe usa um Traje diferente com o fim de que cada peixe seja uma espécie distinta.

Insiram o código seguinte :



Caso a borda for alcançada quando os actores « ressaltarem », seria melhor usar o único código já utilizado até agora do que mandar os actores mudar de direcção (por exemplo apenas girar ). Essa única faixa de código permite que o actor gire ( por exemplo de 180 graus ou da esquerda para a direita) de forma aleatória em determinado intervalo ( por exemplo de 1 a 26).

O uso de



combinado com



situado na pasta **Operadores** significa que esta mudança de direcção vai acontecer uma em cada vinte vezes. Portanto, caso o intervalo for menor ( por exemplo de 1 a 6), o actor girará mais vezes .

O uso do comando potente **SE \_\_\_ ENTÃO** no guião significa que caso certa inscrição entrada for gerada, a reacção indicada correspondente no código acontecerá automaticamente.

O uso do bloco **Repetir sempre** com o bloco de **Se \_\_\_\_\_ Então** garante que esta mudança de orientação do actor será uma acção contínua.

## Exercício

Peçam aos estudantes que testem modificações de « Números aleatórios » e do bloco « Girar de \_\_\_ graus ».

Use o mesmo código que o do actor anterior ( visto acima)

## Adição de algas e de anemonas

Com o fim de dar à programação um ambiente mais autêntico de recife de corais vivos, anemonas, algas ou corais têm de serem adicionados ao palco.

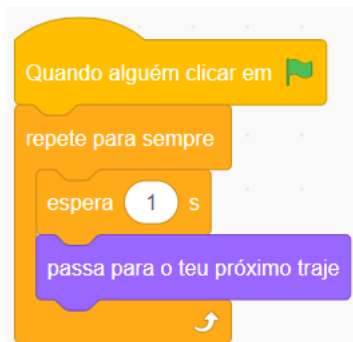


Desenhem um actor de algas clicando no ícone

Use depois a ferramenta pincel a partir do menu na esquerda para criar algas.

Criem duas ou três versões de Trajes para o actor de alga, cada versão com suas fundas em posição diferente para dar um movimento de balanço quando for animada pela codificação.

Insiram a codificação seguinte :

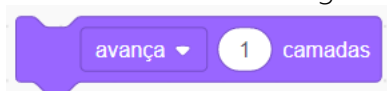


Desenhem um ou dois actores de anemonas.

## Criar um efeito em 3D : Usar 'Plano'

Mas talvez observarem que as algas todas estão colocadas na frente, posição que atrapalha ver os peixes ao mover-se.

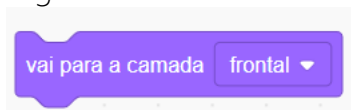
Com o fim de posicionar outra vez algumas algas para trás, o que dá o efeito de o peixe se deslocar na frente e entre plantas individuais, vão até Aparências e seleccionem o bloco seguinte :





Cada plano representa uma zona específica de movimento do actor. Quanto mais elevado for o número, mais actores marinhos nadarão na frente das algas.

Para que uma alga esteja situada na frente do ecrã, usem o bloco de movimento seguinte :



### Quando as criaturas marinhas embaterem

A categoria Sensores leva blocos de código que permitem aos actores interagirem entre eles.

Seleccionem um actor com o código que lhe permita cruzar o caminho de outro actor.

Adicionem o código seguinte para este actor :

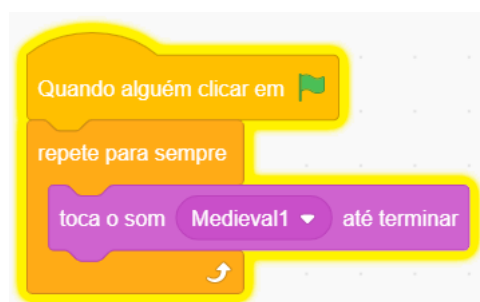


« Remoinho », o efeito especial supracitado e usado no exemplo acima, modificará a forma física do actor. Quanto mais elevado for o número, mais a distorsão será importante. Mas depois de esperar um ou dois segundos, é importante adicionar ao bloco de código o bloco adicional **altera o teu efeito remoinho para 0** ou **cancelar os efeitos gráficos**. No caso contrário, o actor ficará sempre distorcido. O « bobinar » é um dos numerosos efeitos especiais deste bloco de comando situado em Aparências, como está ilustrado abaixo :

Peçam aos estudantes que testem estas opções e que mudem o número em **adiciona o teu efeito remoinho o valor** \_\_\_

### Um som de ambiente para o aquário

Por fim, seleccionem uma música adequada para combinar com o ambiente dos peixes que se deslocam lentamente.



### Exercício

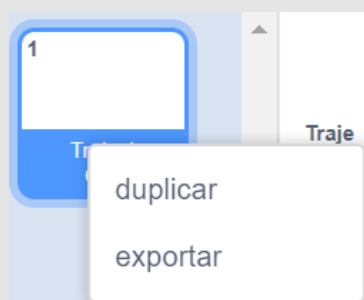
Peçam aos seus alunos desenharem :

- Um palco de floresta ou de floresta com muitos animais movimentando-se como aves, mamíferos e insectos ou
- Uma cena de rua com caminhantes, com pessoas fazendo compras, com ciclistas, com artistas de rua, etc.

### Salvar os actores criados

Os actores criados num projecto podem ser salvos para serem usados em outros projectos.

Posicionar o cursor sobre o actor que vão querer salvar.



Seleccionem exportar este actor.

E escolhem a localização onde vão querer salvar o actor (por exemplo: o desktop, actor Scratch, pasta « Objectos »), dando-lhe nome e salvem-no.

## Lição 14 – A bola-alvo

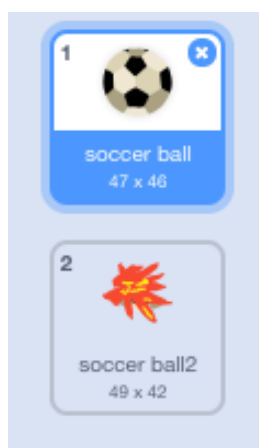
Objectivos da lição :

- Aprender o uso de **variáveis**
- Usar o bloco de código **Número aleatório (Operadores)** para modificar a posição do actor.
- Mostrar como uma acção pode ser programada para acontecer quando alguém clica sobre um actor.

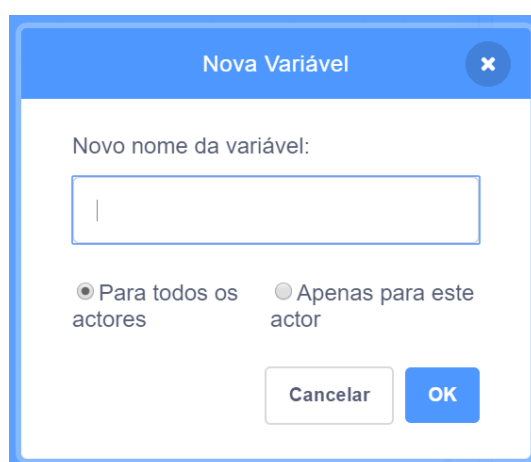
Bola alvo – resumo do plano de codificação

A bola se desloca com uma velocidade aleatória pelo cenário. O utilizador tenta direccionar a bola ao gravar uma pontuação para cada ataque.

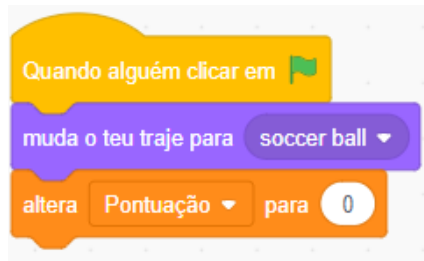
Façam um segundo Traje do actor bola que dê a impressão que ela explode..



Agora vão até a pasta **Variáveis**, seleccionem a opção Criar uma variável.



Digitem Pontuação por baixo da variável e escolhem Para todos os actores. Deslizem depois o bloco seguinte ajustando a pontuação no zero.



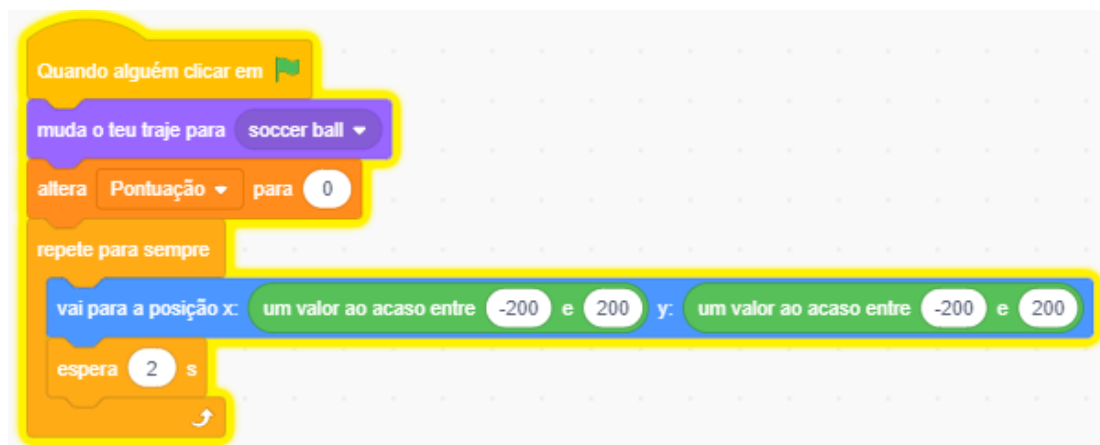
Agora queremos que o actor se desloque pelo palco conforme um modelo definido. Portanto usamos o bloco `um valor ao acaso entre 1 e 0` em Operadores para as coordenadas X (horizontal) et Y (vertical). Os números aleatórios são números que não representam nenhum modelo e que são imprevisíveis. Mas os dois quadrados brancos em `um valor ao acaso entre 1 e 0` dão a possibilidade ao utilizador de definir um intervalo para X e Y. Portanto, neste caso o actor deslocar-se-á por um espaço máximo no palco com o uso do bloco seguinte:



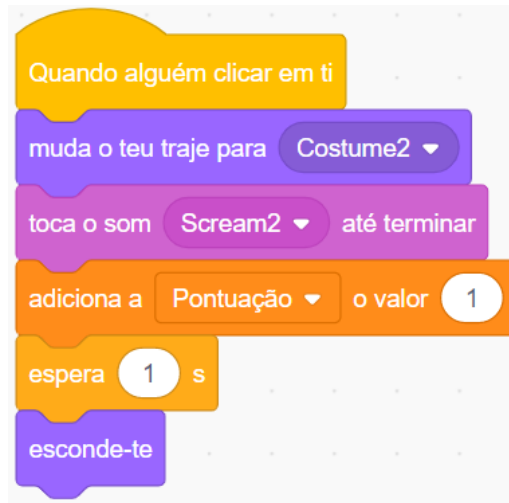
Para certificar-se que o movimento aleatório seja contínuo usem o bloco Repetir sempre.



Mas ele deve ser usado conjuntamente com `espera 1 s`. No caso contrário, a velocidade do gato tornaria difícil o registo de uma pontuação.



Agora, entramos no guião uma faixa de código distinto que dá uma pontuação a cada ataque por bola.



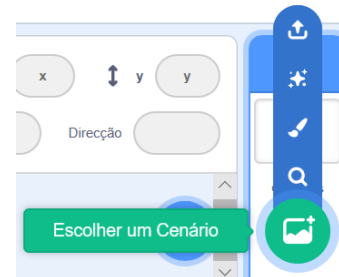
O uso de Mostrar e Esconder (adjacente) significa que a bola desaparece cada vez que tem um ataque. Isso significa, claro, que o jogo é de muito curta duração e que termina quando o actor for atingido uma vez.

Então, para tornar o jogo mais demorado e mais divertido, copiem o actor cinco ou seis vezes. Dêem logo a cada um destes actores de bolas, um pacote distinto de coordenadas X e Y para elas ficarem todas elas em localizações diferentes no início do jogo.

## Lição 15 - Os actores dançantes

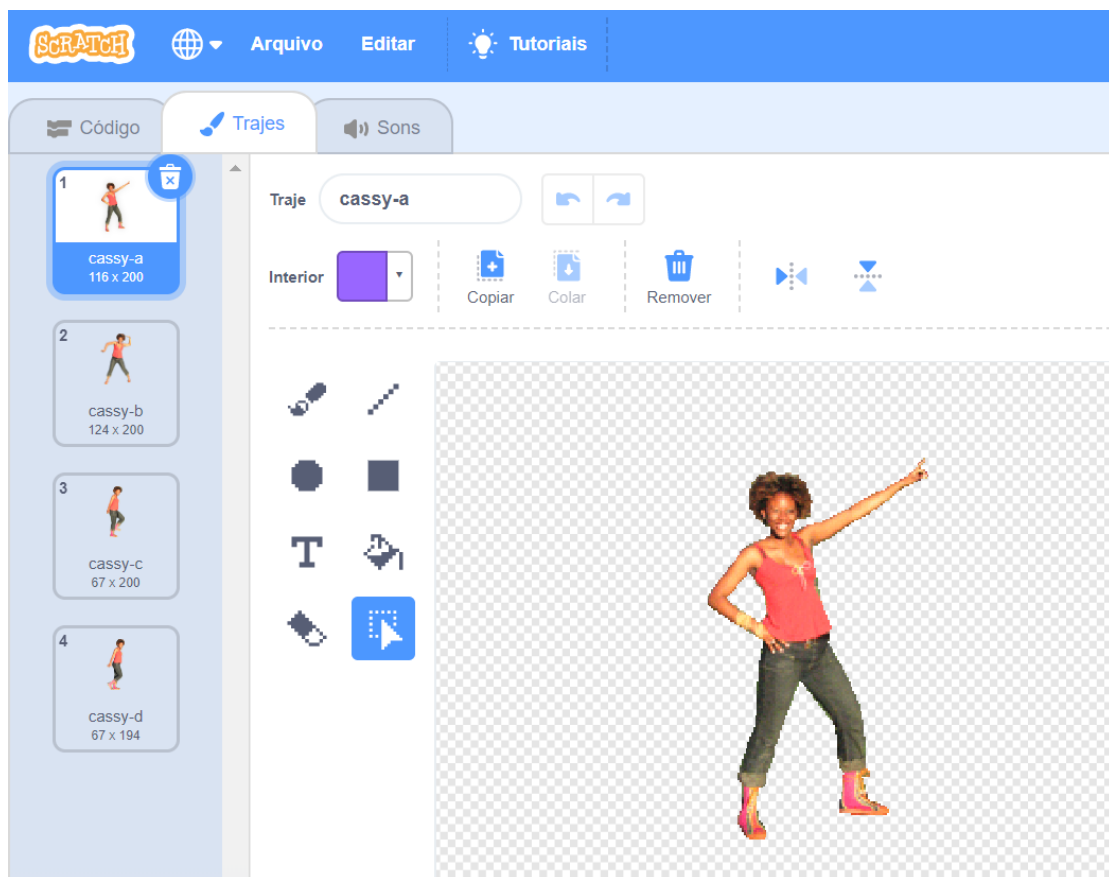
Nesta lição, os estudantes aprendem a criar uma animação de dança por um pacote de comandos a partir das pastas Aparência, Controlo e Som.

Primeiro, cliquem no ícone Palco situado por baixo da secção Palco do ecrã Scratch.



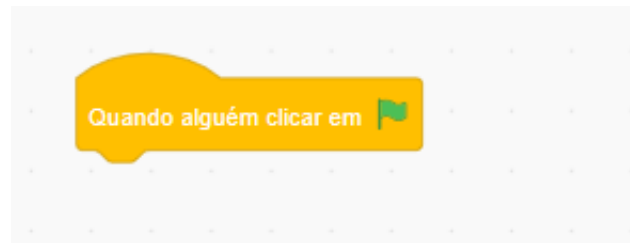
Vão até  e escolhem um cenário adequado para a sequência de dança que vocês vão criar.

Escolhem depois um novo actor que tenha uma escolha para diferentes posturas. Vão até a secção Traje e importem um número de diferentes versões (Trajes) do mesmo actor.



E agora comecemos a animação !

Selecione primeiro

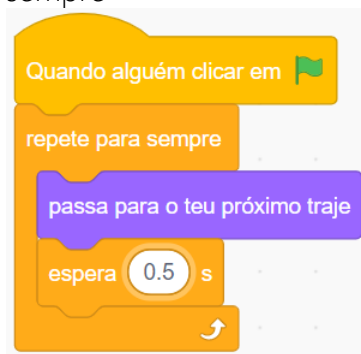


Siga com o bloco Repetir sempre.

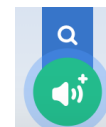
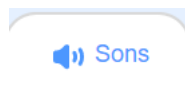
Este comando levará a programação de dança para ser executada constantemente até que o utilizador selecione o ícone **Pare** do círculo vermelho situado na parte superior direita do ecrã Palco.

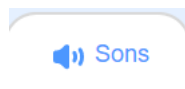
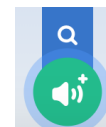
Selecione depois o comando Esperar e reduza a duração de 1 segundo a 0.5 segundo, em caso contrário o actor irá dançar demais lentamente.

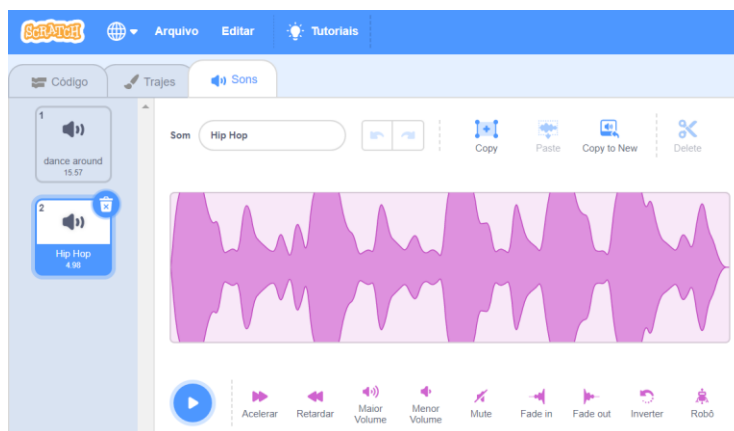
Vão até Aparências, selecione Traje seguinte e insiram-no no bloco Repetir sempre



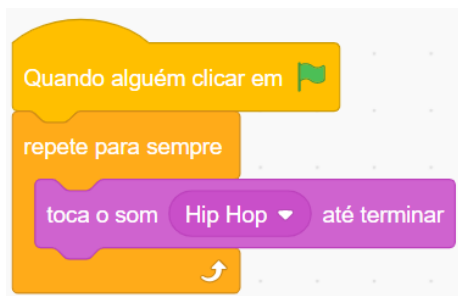
Comecem a programação.



Para adicionar um som, vão até , cliquem no  e selecionem um som adequado para uma dança tal como o Hip-Hop em Repetição.

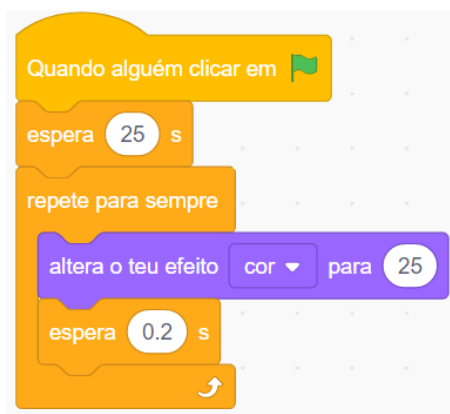


Na secção Actor, adicionar o seguinte :



Comecem a programação .

Agora adicionem o guião seguinte para introduzir um efeito de luz disco no palco.



### Exercício

Peçam aos estudantes que iniciem uma nova programação com um cenário diferente, uma música diferente e três actores dançantes.

Incentivem-nos a desenharem, por exemplo, alguns dançarinos tradicionais em diferentes posturas e a deslocarem ( usar o bloco Esperar) sua aparência no palco. Procurem uma música adequada e criem seu próprio espectáculo musical !



## Lição 16 - Desenhar formas

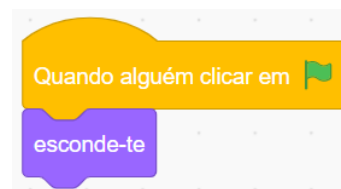
Nesta lição, os estudantes aprendem a entender como desenhar formas geométricas com o uso dum série de guiões.

Eles aprendem também o conceito de Sorte ao usar um bloco de selecção Aleatória a partir de Operadores.

### Criar um guião para desenhar um quadrado

Desta vez, não precisamos dum actor. Mas como todos os comandos de Scratch não podem funcionar sem actor, teremos que escondê-lo.

Use os comandos seguintes para desaparecer o actor:



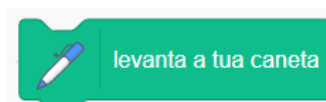
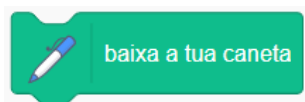
Para desenhar no Scratch, viemos com o cursor no canto inferior esquerdo da interface e seleccionamos a opção Adicionar uma extensão representada pelo ícone.

Cliquem depois na secção Caneta que adicionará automaticamente esta opção a suas categorias temáticas de Bloco.

A partir de Caneta, seleccionem um tamanho e uma cor.

Damos também instruções com caneta para:

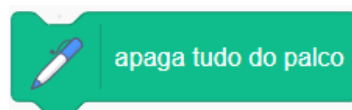
a) começar



e b) acabar de desenhar.

É importante também apagar os desenhos anteriores do palco e voltar a fazer o procedimento de desenho em tela virgem uma vez que a Bandeira verde for seleccionada para começar de novo o projecto.

Portanto use o comando

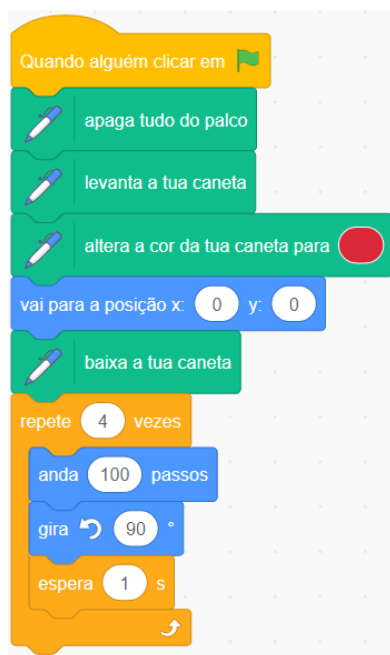


Adicionar o bloco « Esperar 1 segundo » dará ao espectador a possibilidade visual de apreciar o início do desenho e o procedimento efetivo de formação dum novo objecto ao mesmo tempo.

Para certificar-nos que o objeto geométrico que vamos desenhar , ou seja um quadrado, tem dimensões suficientes por uma fácil visualização, usem um número importante de etapas na pasta Movimento, por exemplo : 100 passos.

Para entrar o código no comando que desenhará 4 linhas formando uma caixa, vamos usar primeiro o código Repetir. Para construir um quadrado usa-se **Repetir 4** vezes.

Anotem também que por cada forma geométrica, o ângulo é proporcional ao número de lados, ou seja 360 graus dividido pelo número de lados.  
Por exemplo, um quadrado é 360 dividido por quatro = 90 ;  
Um triângulo é 360 dividido por três = 120 e  
Um círculo é 360 graus dividido por 360 = 1



Então, temos que virar as linhas usando um bloco de graus no comando de Movimento.

Podemos também modificar as cores de cada desenho escolhendo a opção aleatória na categoria Operadores. O espectro de cores vai de 1 a 200. Portanto, escolher uma faixa aleatória elevada, por exemplo 1-200 (ver imagem abaixo) aplicável à variação de cores, vai permitir à programação, a selecção de forma aleatória numa variedade completa de cores em cada execução.

Agora, como podemos fazer um **triângulo** ?



Quantos lados tem um triângulo ?

Quais seriam os ajustes de ângulos (graus) ?

```

Quando alguém clicar em [bandeira]
  esconde-te
  apaga tudo do palco
  levanta a tua caneta
  altera a cor da tua caneta para um valor ao acaso entre 1 e 200
  vai para a posição x: 0 y: 0
  altera a espessura da tua caneta para 1
  altera a tua direcção para 90 °
  baixa a tua caneta
  repete 3 vezes
    gira 120 °
    anda 100 passos
    espera 1 s
  
```

Desenhemos um círculo.

Quantos giros ( graus) há num círculo ?

```

espera 1 s
  
```

Portanto girem um grau por vez. Não usem o bloco `espera 1 s` no guião. Com efeito, seu uso significará que o círculo precisará de 360 segundos para acabar.

Adicionem o código seguinte na repetição :

```

adiciona a o matiz da tua caneta o valor 10
  
```

```

repete 360 vezes
  gira 1 °
  anda 1 passos
  
```

Para ter: Podem apreciar o efeito colorido.

```

repete 360 vezes
  adiciona a o matiz da tua caneta o valor 10
  gira 1 °
  anda 1 passos
  
```

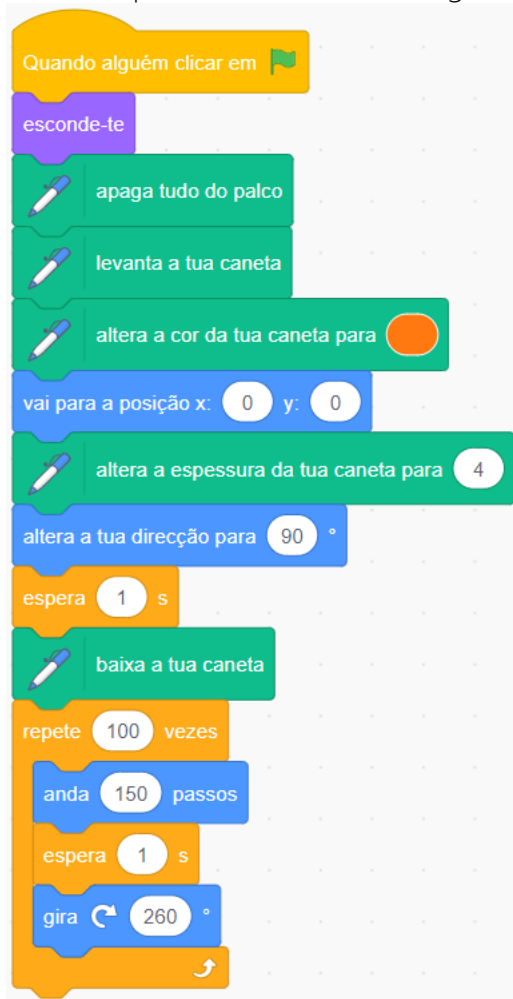
## Exercício

1. Desenhem um pentágono
2. Escrevam uma programação que desenha três formas diferentes que aparecem no cenário em distintos momentos e em distintos lugares.

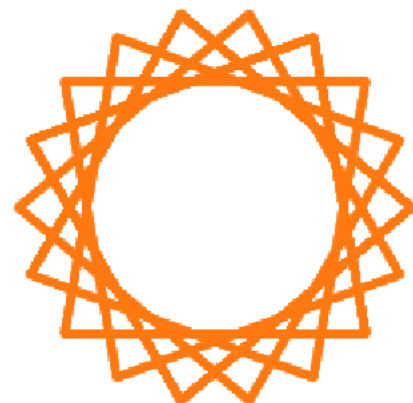
## Outras Formas

Testem com diferentes ângulos e « Repetir » na programação.



Por exemplo, seleccionem o código abaixo e admirem os resultados.

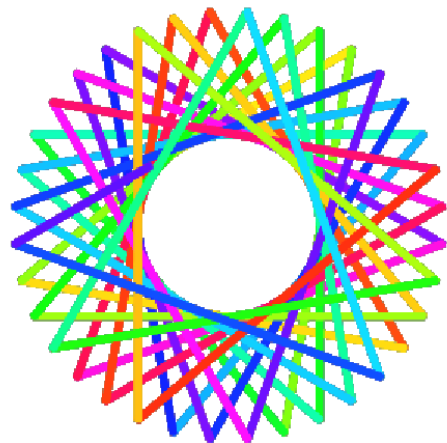


```
Quando alguém clicar em [ ]
  esconde-te
  apaga tudo do palco
  levanta a tua caneta
  altera a cor da tua caneta para [orange]
  vai para a posição x: 0 y: 0
  altera a espessura da tua caneta para 4
  altera a tua direcção para 90 °
  espera 1 s
  baixa a tua caneta
  repete 100 vezes
    anda 150 passos
    espera 1 s
    gira 260 °
```

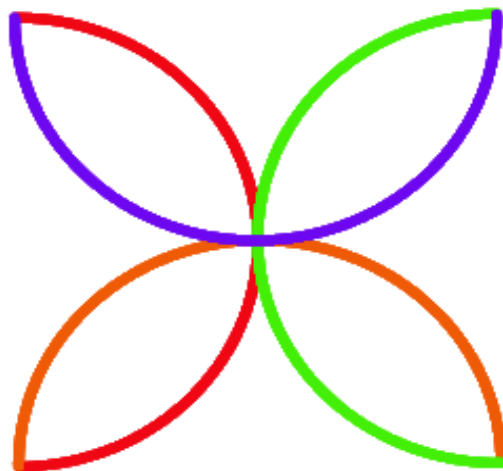


Insiram o código seguinte :

```
Quando alguém clicar em   
vai para a posição x: 0 y: 0  
altera a espessura da tua caneta para 5  
esconde-te  
apaga tudo do palco  
levanta a tua caneta  
espera 1 s  
baixa a tua caneta  
altera a cor da tua caneta para   
repete 360 vezes  
anda 150 passos  
adiciona a o matiz da tua caneta o valor 10  
gira 130 °  
anda 100 passos
```



Criam uma flor de quatro pétalas



Obtém-se este efeito ao desenhar quatro meio-círculos (180 graus) e no fim de cada um ao virar na direita (90 graus).

```
Quando alguém clicar em [bandeira]
  esconde-te

Quando alguém clicar em [bandeira]
  vai para a posição x: 0 y: 0
  altera a tua direcção para 90°
  apaga tudo do palco
  repete 4 vezes
    repete 180 vezes
      anda 2 passos
      gira 1°
    altera a espessura da tua caneta para 5
    baixa a tua caneta
    adiciona a o matiz da tua caneta o valor um valor ao acaso entre 1 e 200
    gira 90°
```

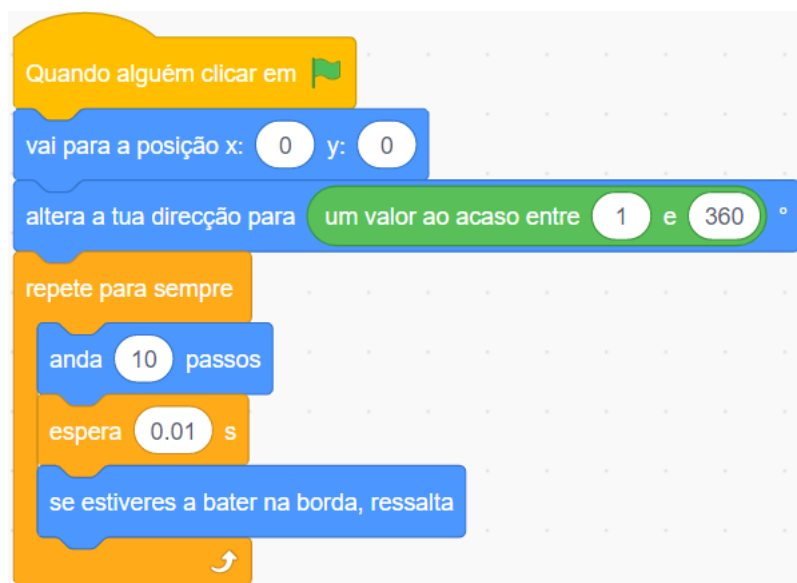
## Lição 17 – A bola que ressalta

Nesta lição, os estudantes vão acostumar-se com os **Graus** e com as **Variáveis do espaço**.

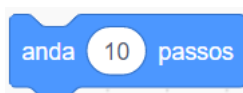
Desenhem primeiro uma Bola com o uso da opção Desenhar um novo actor.



E escrevam o guião seguinte :



Testem a velocidade da bola ao aumentar e ao diminuir o número na zona do bloco Andar \_ passos:



Podemos também controlar a velocidade da bola durante o jogo ao usar a categoria **Variáveis**.

Seleccionem Criar uma variável (para todos os actores) e digitem a palavra Velocidade na opção do Nome de variável

Um bloco Velocidade aparece no cenário



Cliquem com o botão direito sobre o ícone (se usarem o rato) e seleccionem a opção Cursor, que ajuda o usuário a ajustar a velocidade da bola em movimento.



Removam o bloco Velocidade da categoria Variáveis e deslizem-no na caixa branca (número) do bloco **andar \_\_\_\_\_ passos**.

Agora, o código de guião lê-se da forma seguinte :

```
Quando alguém clicar em [bandeira]
vai para a posição x: 0 y: 0
altera a tua direcção para um valor ao acaso entre 1 e 360 °
repete para sempre
  anda Velocidade passos
  espera 0.01 s
  se estiveres a bater na borda, ressalta
```

Gerar uma cauda de cometa na bola que ressalta

```
Quando alguém clicar em [bandeira]
esconde-te
vai para a posição x: 0 y: 0
apaga tudo do palco
levanta a tua caneta
espera 1 s
mostra-te
baixa a tua caneta
adiciona 10 à espessura da tua caneta
repete para sempre
  altera a tua direcção para um valor ao acaso entre 1 e 360 °
  se estiveres a bater na borda, ressalta
  anda Velocidade passos
```



## Exercício

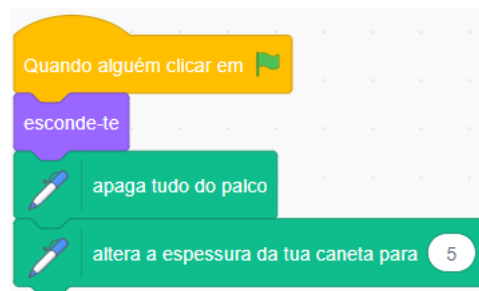
1. Escrevam uma programação com várias bolas que ressaltam.
2. Escrevam uma programação com a temática do mar que apresenta algum número de criaturas marinhas diferentes que ressaltam, tais como medusas e tubarões.

## Lição 18 – Desenhar à mão livre

*Nesta lição, os estudantes aprendem a criar um programação dando ao utilizador a possibilidade de desenhar à mão livre.*

O utilizador deve primeiro, como em todas as programações de desenho de formas, desaparecer o gato (ocultar). Com efeito, seria muito difícil desenhar corretamente com um actor tal como um gato usado como ferramenta de desenho.

Depois o utilizador usa o bloco vazio e logo o bloco de afiador de caneta definido para ter uma caneta de desenho de tamanho suficiente.

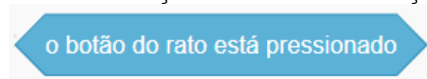


O utilizador pode tornar-se a caneta de desenho codificando o movimento do rato, como já fizemos em lição anterior quando o movimento dum actor animal era controlado pela acção do rato.

Mas temos agora de certificar-nos que a programação considera o fato que o movimento do utilizador não é contínuo. No caso contrário, o ecrã iria se encher de uma linha sem fim.

Como consequência, temos de codificar dentro de um pacote de instruções para que o artista possa levantar sua caneta em qualquer momento para acabar de desenhar e para continuar a desenhar em outro lugar do ecrã.

Conseguimos esta acção com a codificação num guião principalmente baseado no



bloco

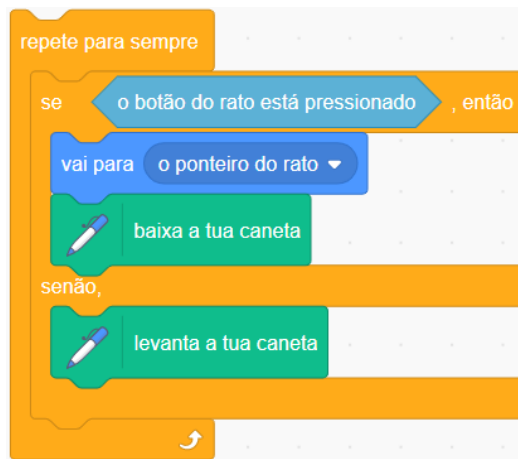
Da pasta Sensores

E



da pasta Controlos.

O que dará:



O guião final lerá da forma seguinte:



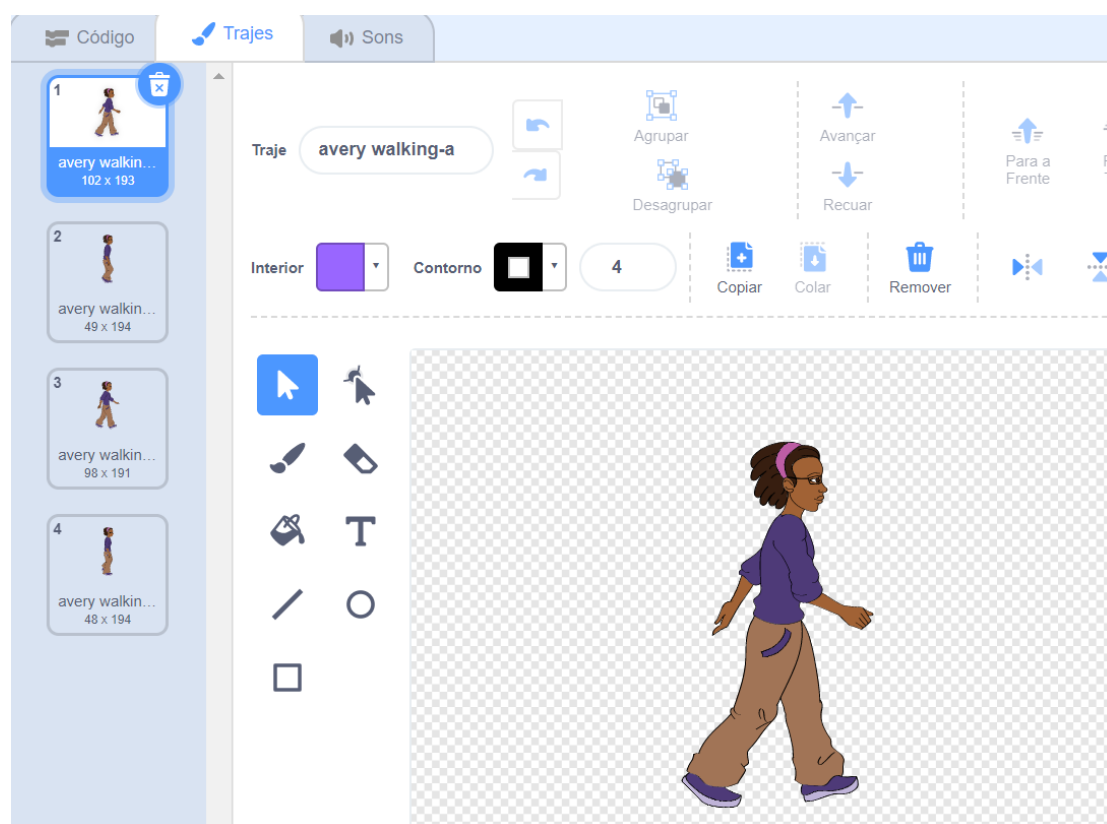
## Lição 19 – Levar o cão a passear

Nesta lição, está apresentada aos alunos a funcionalidade de Temporizador. Com essa funcionalidade, é possível modificar os cenários e os actores em determinados momentos específicos.

Os alunos usarão o bloco « Apontar em direcção » que controla o rumo pelo qual o actor olha e se desloca.

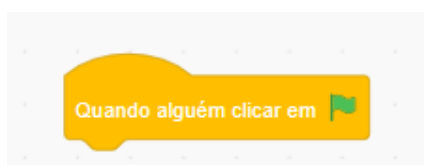
A lição baseia-se numa menina ou num menino que leva um cão a passear por várias paisagens.

Seleccionem na secção Pessoas da biblioteca dos actores um que seja adequado com vários Trajes de caminhada.



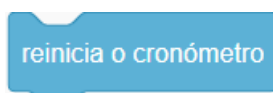
Vão até a biblioteca de cenários da opção Palco e seleccionem três palcos diferentes na pasta Fora.

Pelo comando



Acrescentamos dois blocos adicionais para cima e por cima daqueles que o utilizador já conheceu nas lições anteriores.

Escolhem primeiro o bloco



a partir da pasta **Sensores**, o que inicializa o tempo em zero quando clicar na Bandeira verde.

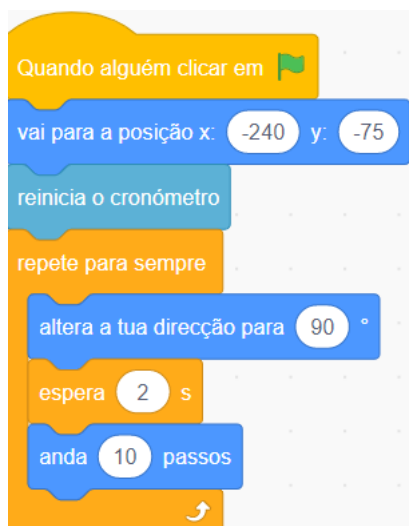


A segunda característica adicional é o bloco



que codifica a direcção do actor . Isso é um dado importante pois o roteiro incluirá a meia-volta da menina no fim do passeio.

Assim, o guião inicial seria:



Colocamos agora por baixo dos blocos mostrados acima, um indicador adicional que vai voltar a posicionar o actor na extremidade esquerda ( fim do ecrã) ao longo do eixo X quando ele atingir a extremidade direita do ecrã. Como queremos que a menina seja vista passeando em diferentes cenários (de um palco para o outro) como se fosse constantemente um passeio demorado.

Para realizá-lo ,usamos um bloco de código que permite acontecer alguma coisa quando se alcança o fim do ecrã.

É o bloco:



Vão até a pasta **Operadores** e seleccionem o bloco **superior** a

que será situado no espaçamento do bloco no guião mostrado acima.

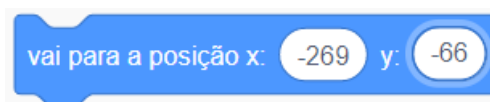
Digitem os números 226 no quadrado direito que representa o número de pixels (eixo dos Y) na extremidade direita do ecrã.



Então, entrem **o x da tua posição**

a partir de **Movimento** na caixa esquerda de

Usamos o **o x da tua posição** como o actor que se move apenas no eixo horizontal (X).



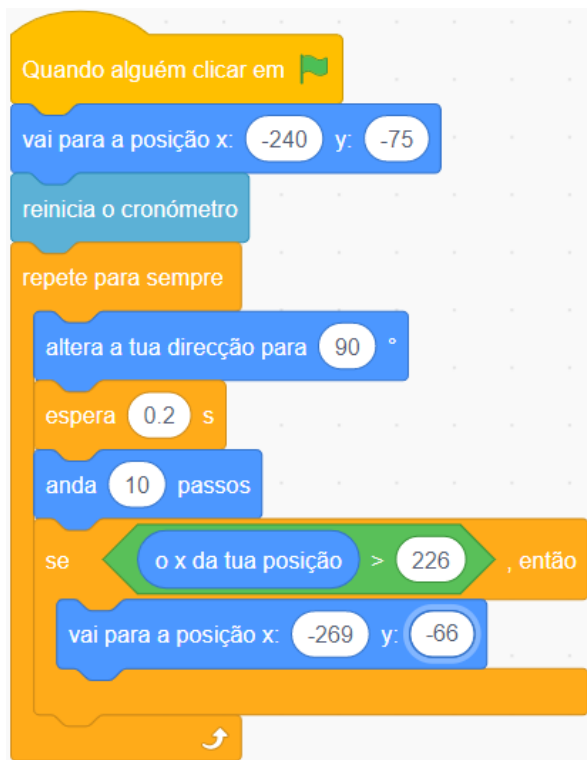
A partir da mesma pasta peguem



E insiram no bloco

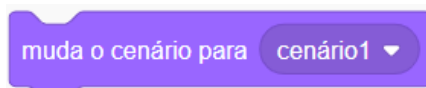
Observem que a coordenada X tem um número negativo elevado que, quando a programação começou, deixa a impressão que a menina está a caminhar no palco (ecrã).

O código vai aparecer agora da forma seguinte :

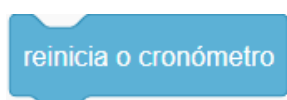


Vamos agora passar ao guião Palco no qual vamos codificar em três cenários que mudarão conforme um controlo de parâmetro de tempo.

Os comandos principais para estes blocos são :



(Aparências)



(Sensores)

e associamos



(Operadores)

Para ter :

```
Quando alguém clicar em [bandeira verde]
  muda o cenário para Desert
  repete para sempre
    se o valor do cronómetro > 11 e o valor do cronómetro < 22, então
      muda o cenário para Winter
    se o valor do cronómetro > 23 e o valor do cronómetro < 34, então
      muda o cenário para Savanna
```

Observem que os números entrados são baseados no tempo necessário para a menina se deslocar no eixo dos X de um lado ao outro do ecrã.

É óbvio que é fundamental que o codificador possa sincronizar o cronómetro com o reposicionamento do actor na parte esquerda do ecrã cada vez que ela atingir a borda do ecrã. Isso provavelmente levará a algumas « tentativas » e a alguns « erros » por parte do estudante.

Então, para completar esta parte da programação, vamos acrescentar uma faixa de código adicional que limitará o tempo de a menina avançar para a direita, o que facilitará uma meia-volta quando ela chegar no fim do último cenário e recuar de alguns passos ( na esquerda).



Primeiro, vamos medir o tempo necessário para o actor caminhar no início do primeiro palco até ao último palco. Neste projecto específico, 34 segundos foram inseridos no Cronómetro Se \_\_\_\_então

```
se o valor do cronómetro < 34, então
```

e o colocamos no código da seguinte forma :



```

Quando alguém clicar em [bandeira]
vai para a posição x: -240 y: -75
reinicia o cronómetro
repete para sempre
  se o valor do cronómetro < -34, então
    altera a tua direcção para 90°
    passa para o teu próximo traje
    espera 0.2 s
    anda 10 passos
    se o x da tua posição > 226, então
      vai para a posição x: -269 y: -66
  
```

Agora precisamos de um bloco adicional que vai inverter a direcção do actor que caminha o que levará ao fim da programação:

```

Quando alguém clicar em [bandeira]
repete para sempre
  se o valor do cronómetro < 34, então
    altera a tua direcção para 90°
    passa para o teu próximo traje
    espera 0.2 s
    anda 10 passos
    se o x da tua posição > 223, então
      vai para a posição x: -269 y: -66
  espera 6 s
  pára tudo
  
```

Agora, vamos introduzir um segundo actor, ou seja um cão que late.



Entrem um código similar àquele que foi usado no guião da menina.

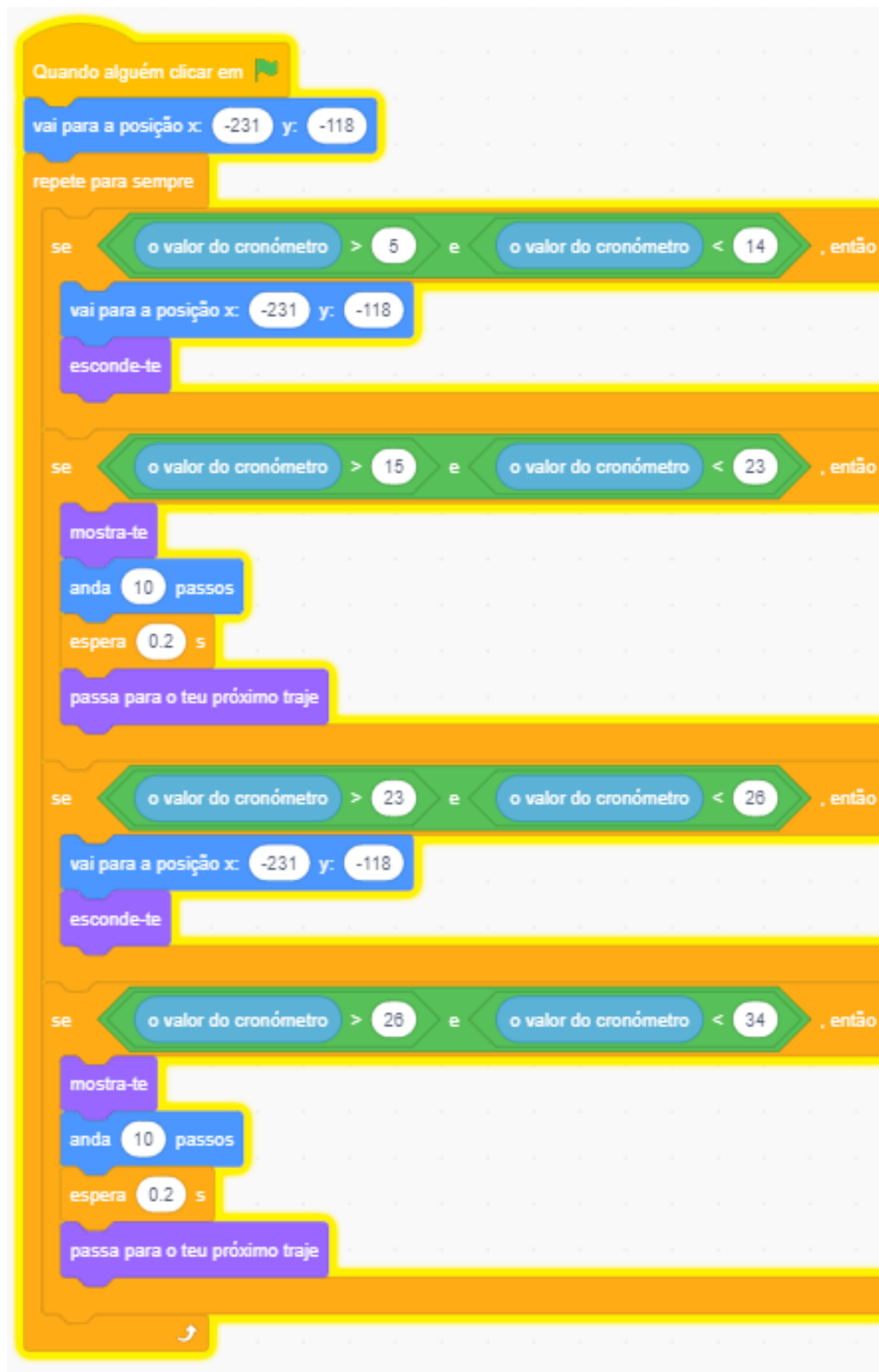
Por outro lado, organizaremos o aparecimento do cão depois de a menina ter chamado seu animal de estimação para passear com ela.

Desta forma, como o animal aparece alguns segundos depois de a menina começar a passear e devido ao seu convite a juntar-se a ela e com alguns passos a caminhar atrás da sua dona, temos de codificar no guião um/ alguns comando(s) que esconderá o cão antes de ele alcançar o fim do ecrã (observem que os cenários mudam no mesmo momento em que a menina alcança o fim do ecrã). O cão apenas volta a aparecer quando a menina tem avançado alguns passos.

Desta forma, usamos a opção **Mostrar** e **Esconder** da pasta **Aparências**.

```
Quando alguém clicar em [ ]
vai para a posição x: -231 y: -118
repete para sempre
  se o valor do cronómetro < 5, então
    esconde-te
    espera 3 s
    anda 10 passos
    passa para o teu próximo traje
  se o valor do cronómetro > 5 e o valor do cronómetro < 11, então
    mostra-te
    anda 10 passos
    espera 0.2 s
    passa para o teu próximo traje
```

Como para o guião do actor da menina, uma vez que o fim do último ecrã for alcançado, entrem um código que inverte a direcção do actor para ele fazer alguns passos para trás. A codificação dos parâmetros do cronómetro para o cão deve sincronizar-se com a do guião da menina.



Por fim, teríamos também de inserir um código para permitir um pouco de conversa entre a menina e o cão.

## Lição 20 – Planeamento e concepção dum jogo<sup>i</sup>

Os alunos têm de ser informados da importância de **planejar** primeiro o jogo com um lápis, uma caneta e papel antes de apressar-se para escrever os guiões. Eles devem entender a necessidade de definir a meta do jogo, como alcançá-lo e o papel de cada elemento do jogo tal como os diferentes actores e como eles interagem (efeito).

O plano escrito (ou Algoritmo) é logo usado para construir o jogo.

Mas, o tutor deveria primeiro providenciar um exemplo dos diferentes tipos de jogos que podem servir de pontos de partida ou modelos às criações dos alunos.

Exemplo de **orientações** de jogo « Ataque de tubarão ! »

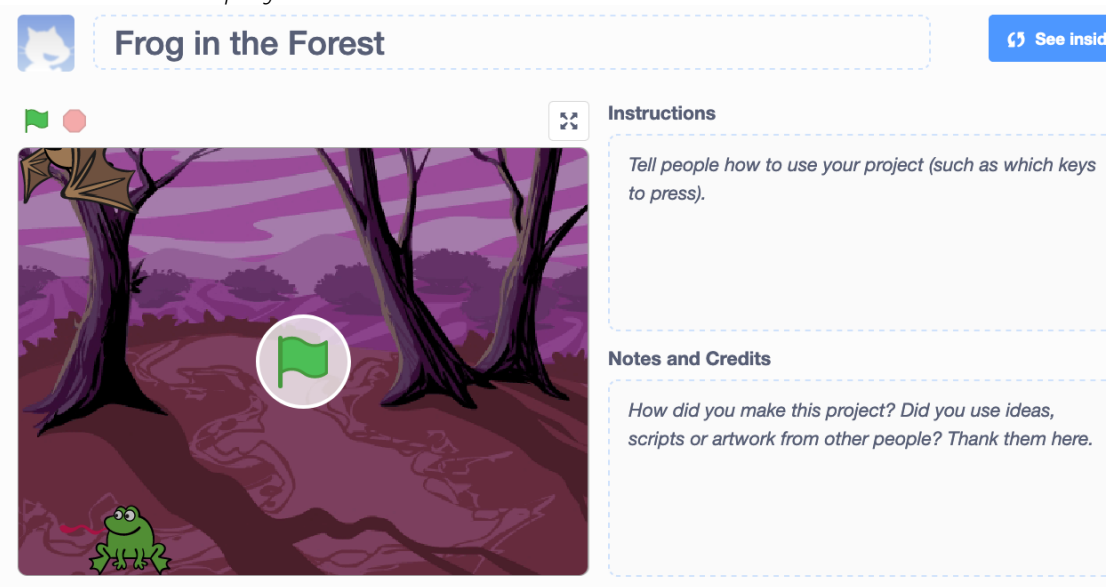
- *Quantos personagens (dois actores para começar)*
- *Papel dos personagens (actor Peixinho, actor Tubarão mau)*
- *Operações dos personagens (o movimento dum actor é controlado pelas quatro « teclas seta » ; o segundo personagem « seguirá » o primeiro actor usando o bloco de código « orientar-se para »)*
- *Qual cenário ficaria melhor com o tipo de jogo que estou a criar ?*

### Descarregar um jogo no site Scratch

Alguns exemplos de projectos de todos os alunos deveriam ser baixados na conta on-line Scratch da escola, da turma ou do grupo que vocês ajudaram na configuração no site de Scratch, incluindo os melhores projectos armazenados também na sua própria conta de projecto Scratch.

É importante então que seja proporcionada uma nota explicativa, curta e bem escrita, explicando a temática e o funcionamento do projecto, e isso para cada projecto baixado dando benefício aos utilizadores on-line.

Uma vez baixado seu projecto, vão até a página do projecto e preencham as secções na direita do ecrã intituladas *Instruções*, *Notas e créditos* e *Acrescentar sinalizadores de projecto*.



The screenshot shows the Scratch project page for 'Frog in the Forest'. The main area displays a forest scene with a green frog in the bottom left corner and a green flag in the center. To the right, there are three sections: 'Instructions' with a text box containing the text 'Tell people how to use your project (such as which keys to press).', 'Notes and Credits' with a text box containing the text 'How did you make this project? Did you use ideas, scripts or artwork from other people? Thank them here.', and a 'See inside' button at the top right.

## Lição 21 - Jogo – Ataque de tubarão !

Esta lição apresenta aos alunos a) a programação das **Teclas de seta** para controlar o movimento dos actores (característica frequente de numerosos jogos de computador) e o b) Código **Pare tudo**.

Esta lição apresenta também um comando potente **Toca nele** e o comando **se** (com seu impacto de causa para efeito ).

### Condições de realização do jogo – Resumo do plano de codificação

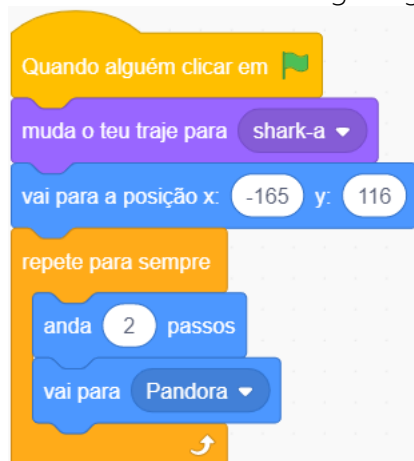
O tubarão está perseguindo o peixinho, que tenta desesperadamente escapar. Quando o tubarão o atinge, o peixinho desaparece como se tivesse sido engolido pelo tubarão que troca seu Traje para fingir um procedimento de deglutição. O tubarão diz com alegria « Miam ! » e o jogo termina.

Seleccionem a partir da pasta Animal a) o actor Tubarão com duas mudanças de Trajes ( boca « aberta » e « fechada ») e b) o actor dum peixinho.

Dêem nomes ao tubarão e ao peixinho. Neste exemplo dei o nome de Kraken ao tubarão e de Pandora ao peixe.

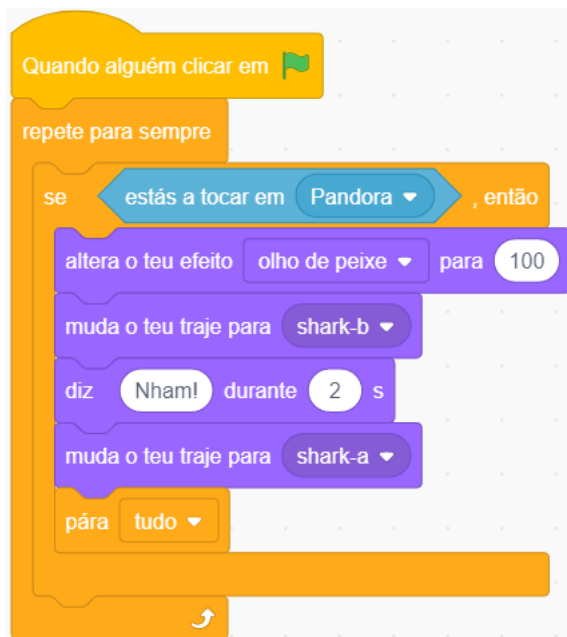


Entrem as faixas de código seguintes no guião de Kraken ( tubarão ) :



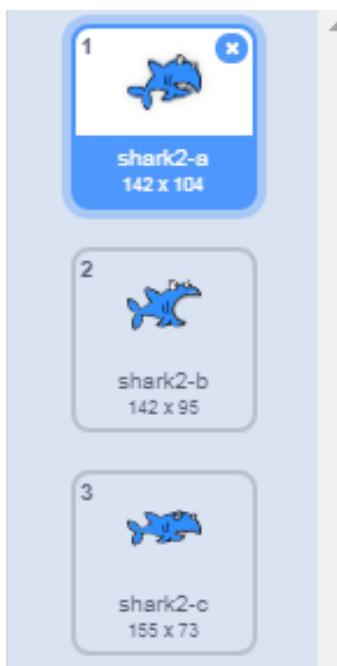
O bloco de código da repetição *repetir sempre* o conteúdo que contém significarão que, o tubarão *sempre seguirá* qualquer que seja o lugar onde ficar Pandora.

O « andar 2 passos » deveria ser uma velocidade suficiente para a primeira versão deste jogo Ataque de tubarão pois permitirá ao peixinho de escapar de Kraken.



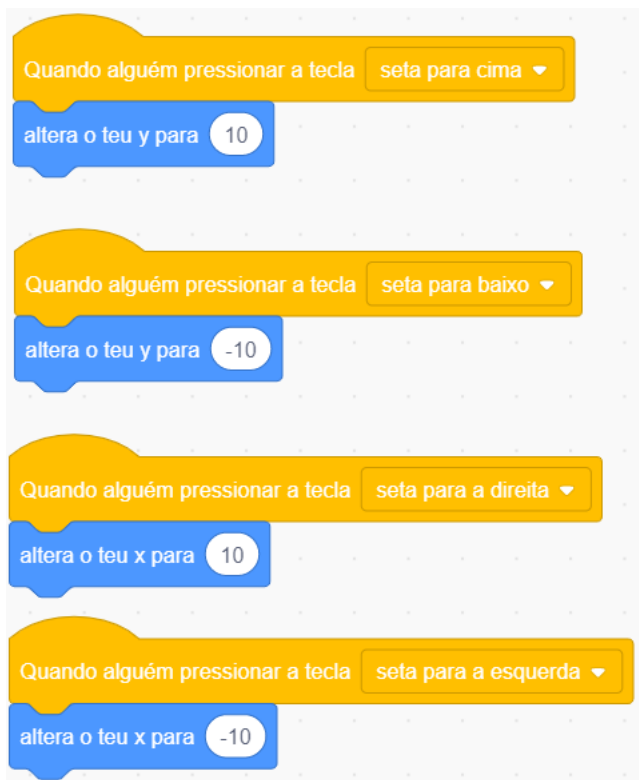
Na faixa de código acima, a opção « tocar » está na pasta **Sensores** .

As faixas de código adicionais da pasta **Aparências** darão a impressão que o Tubarão come o peixinho alternando o Traje com as opções de boca a) fechada b) aberta (Trajes).

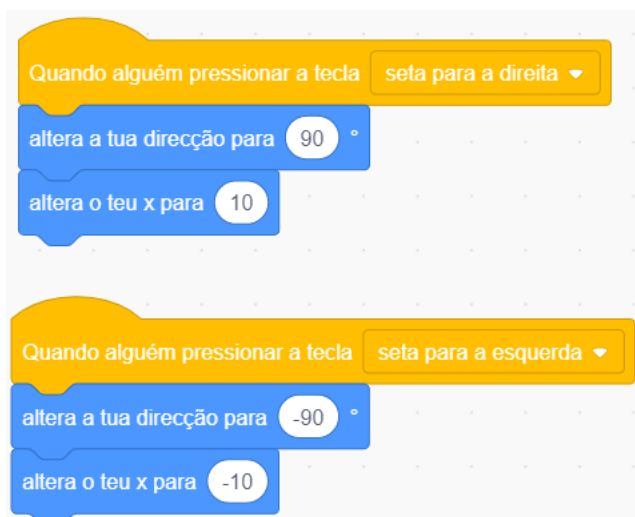
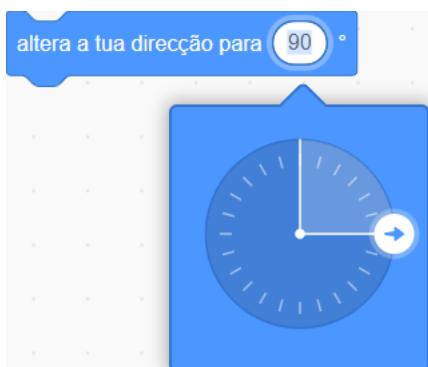


### O guião de Pandora (peixinho)

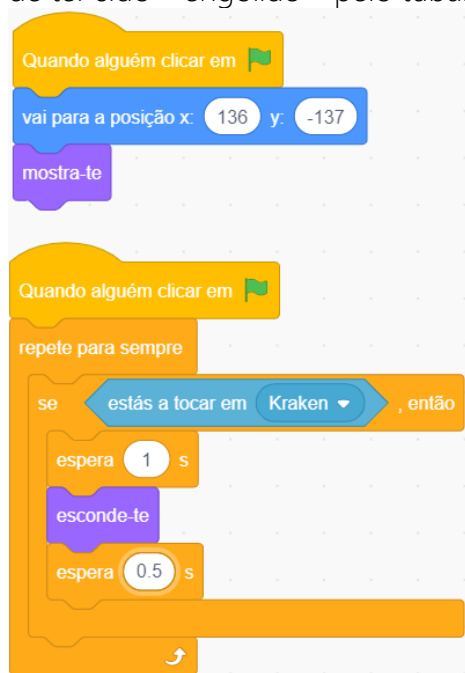
Digitem as faixas de código seguinte no guião de Pandora (peixe) :



Para certificar-se que o peixe irá sempre na mesma direcção quando ele é movido com as teclas de seta esquerda e direita, o bloco **dirigir-se na direcção de**, da categoria Movimento, deve ser adicionado ao guião como na figura seguinte :



Por fim, inserir os blocos *Mostrar e Esconder*, assim como duas opções *Esperar* (conforme mostra a figura abaixo) , significará que Pandora desaparecerá depois de ter sido « engolido » pelo tubarão.



Como podem reparar com o código acima mostrado ,o jogo « Ataque de Tubarão ! » terminará com o o uso do *Pare Tudo*.

## Tubarões ameaçados



Existem provas de fósseis de tubarões que existiram há 400 milhões de anos atrás, o que é admirável quando sabemos que os primeiros dinossauros apareceram no planeta apenas 243 milhões de anos atrás.

Conhecemos hoje mais de 1000 espécies de tubarões e descobrem-se novas espécies cada ano.

Mas a existência deste admiráveis peixes é ameaçada pelas atividades do homem. Avaliam-se em 70 milhões de tubarões mortos cada ano, e numerosas espécies estão ameaçadas agora e poderiam desaparecer daqui até próximos prazos.

Salvar estes animais é fundamental pela recuperação da vida dos oceanos na Terra.

### Exercício

#### Perseguição de carros

Peçam aos alunos usarem os comandos e os procedimentos de codificação ensinados no « Ataque deTubarão ! ». Lição dedicada à criação de um projecto de perseguir carros .

Africa Code Week é uma iniciativa da SAP. O conteúdo inteiro desenvolvido por Brendan Smith a favor do Camden Education Trust, é propriedade intelectual da Africa Code Week.



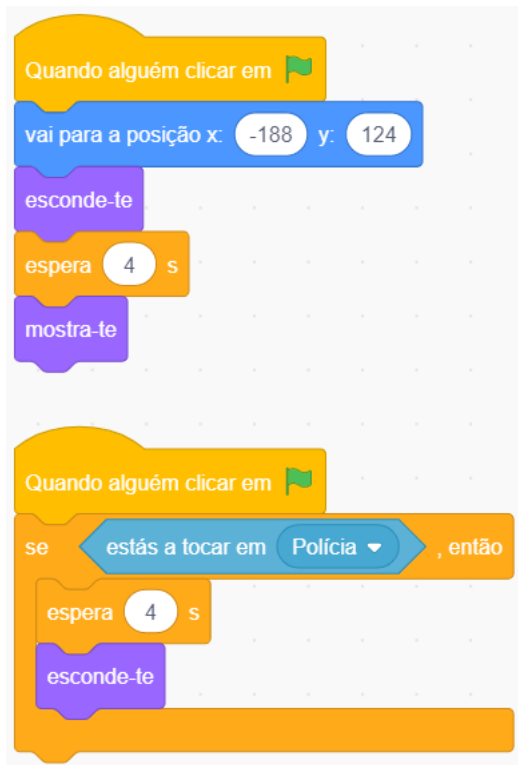
## Código para um jogo de perseguir carros

Peçam primeiro aos alunos usarem o **Editor de desenho** para desenhar um carro de polícia e um carro de ladrão, assim como uma série de cenários de estrada adequados.

Peçam aos alunos usarem um código similar àquele abaixo mostrado para o carro de polícia :



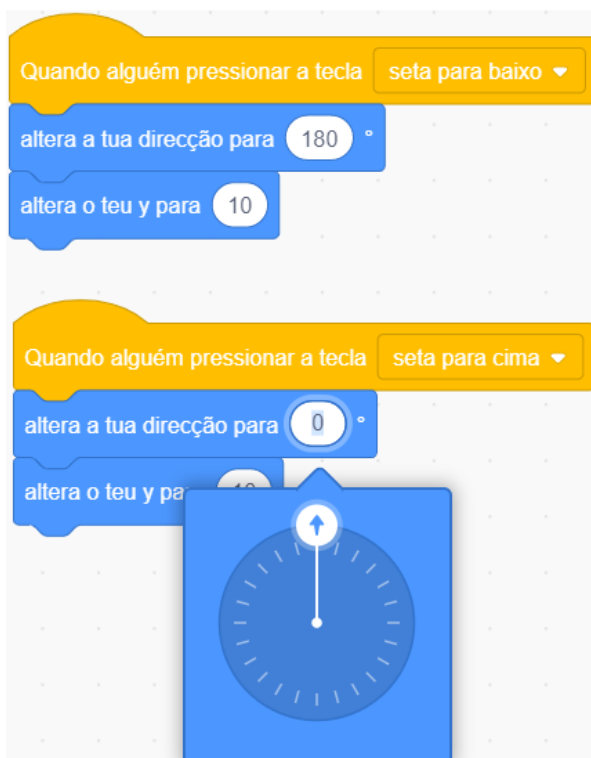
Peçam aos alunos usarem um código similar àquele abaixo mostrado para o carro de ladrão :



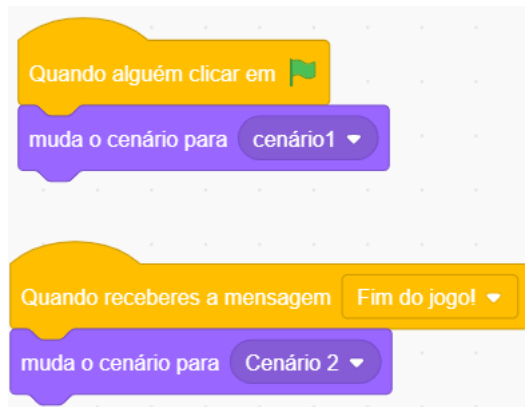
Observação : programar as quatro teclas de seta para controlar a direcção e o movimento do carro

Em « Ataque de tubarão ! » blocos de códigos específicos eram usados para que o peixinho caçado fosse girado para esquerda quando ele se movia para a direita. Desta vez, devemos também estar com o carro perseguido para cima (norte) e para baixo (sul) quando vamos para estas estas direcções.

Vejam figura abaixo:



Programação em diferentes Cenários:



## Lição 22 - Ataque de tubarão avançado

*Esta lição aumenta o grau de complexidade do jogo "Ataque de tubarão !" ao introduzir um elemento competitivo através dum **cronómetro** no projecto (comando Sensores) da **secção aleatória** (noção de acaso), através do uso de variáveis simples na forma de «vidas» e através da criação de diferentes **níveis**.*

Observação: nas matemáticas, uma **variável** é um valor que pode mudar no âmbito dum problema ou de um conjunto dado de operações. Neste exemplo, usamos variáveis para representar o número de vidas dum dos actores, que muda depois de uma interacção com outro actor.

### Condições de realização do jogo – resumo do plano de codificação

O tubarão persegue um peixinho. Este peixinho tem três vidas e deve ficar vivo durante 30 segundos ,depois o jogo termina automaticamente. Cada vez que o tubarão tocar o peixinho, o peixinho perde uma vida.

Depois de dez segundos, o jogo passa para um nível diferente que gera umas modificações no cenário : uma velocidade aumentada do tubarão e o surgimento dum caranguejo que, se tocar o peixinho levará a reduzir a vida do peixinho.

Se o peixinho ainda está vivo depois de 30 segundos, ele ganhou.

Como na lição anterior, (Ataque de tubarão!), vamos usar um Tubarão (Kraken), um peixinho (Pandora) e um cenário (Underwater) da pasta *Fora* de Scratch.

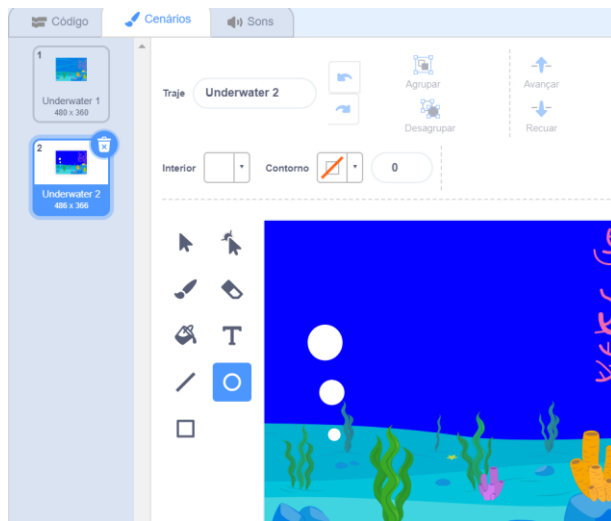
Agora, comecem por descarregar o projecto anterior *Ataque de tubarão !*

Mas desta vez, vamos modificar de forma significativa os guiões do tubarão e o do peixinho ; deslizem distintas versões de cenário *Underwater* que representarão graus de dificuldade crescente e registrem as mensagens *Você ganhou!* e *Você perdeu !* para indicar o fim do jogo.

### **Comecem**

Cliquem no ícone *Cenário*, e dupliquem uma vez o cenário *Underwater*.

Modifiquem este novo nível (nível 2) incluindo funcionalidades adicionais tais como cerdas/ hastes adicionais ou maiores nas algas, corais e rochedos, assim como uma mudança de cores de cenário em azul mais escuro.



Uma vez a modificação terminada, dupliquem o nível 2 duas vezes a mais.  
 Com o segundo novo nível (nível3), usem a opção de edição para entrar no *Editor de desenho*. E digitem as frases adequadas em impressões grandes em negrito tais como *Você perdeu !* usando o ícone T (exto).  
 Com o terceiro novo nível (nível 4), digitem o texto *Você ganhou !* com impressões grandes em negrito.  
 Para voltar a posicionar o texto em uma localização adequada, cliquem no ícone T(exto). Coloquem o cursor no pequeno rectângulo preto que aparece no canto superior esquerda do texto no ecrã ( mostrado abaixo). Um ícone « mão » vai aparecer e permitirá ao operador deslizar o texto até uma nova localização.



### Codificação dos níveis adicionais no guião

Para codificar os três níveis adicionais na programação, vamos primeiro até ao guião do actor de peixinho.  
 Agora coloquem um *cronómetro* que permitirá mudar os cenários conforme um determinado intervalo.

Vão à pasta *Controlo* e escolhem :

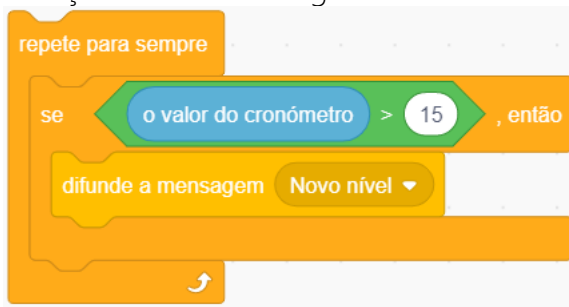


Agora vão à pasta « *Operadores* » e ao bloco de código « *maior que* »



No primeiro quadrado branco vazio acima mostrado, coloquem o **Cronómetro** a partir da pasta *Sensores*.

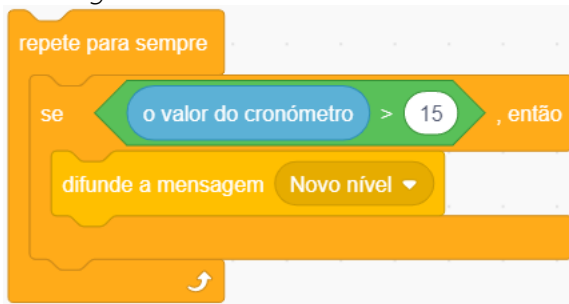
No segundo quadrado branco vazio, insiram um número que representará uma duração medida em segundos :



Nos blocos *Repetir para sempre se*, acima mostrado, coloquem a opção *enviar a todos* a partir da pasta *Eventos*.

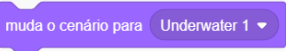


Para certificar-se que o *cenário* mude após 15 segundos, seleccionem *Nova mensagem* no bloco « *Enviar a todos* » e escrevam « *Novo nível* ».

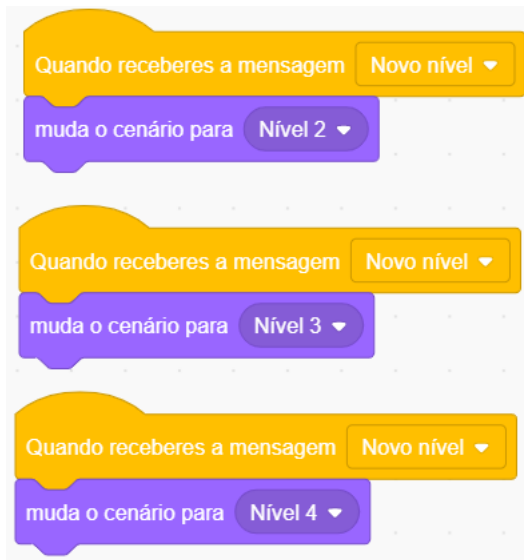


**Enviar a todos** é um código de comando muito importante em Scratch pois envia uma mensagem para outra parte da programação que solicita aplicar uma modificação.

Mas para funcionar, tem de ter um comando correspondente **Eu recebo** que, neste caso , será posicionado no guião do **cenário**.

*Observação :Existe uma alternativa e uma maneira mais simples de mudar os planos de fundo, ou seja o uso direto de  no actor melhor do que no guião do cenário. Entretanto, os alunos estão incentivados a acostumarem-se com a importância dos comandos « *Enviar a todos* ».*

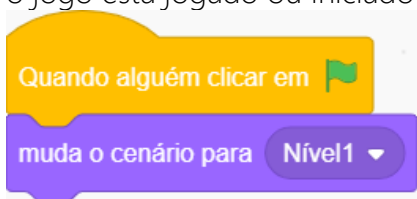
Então, vão até o guião do cenário. Entrem o comando **Quando eu recebo**\_\_\_\_ e logo a faixa de código **alternar no cenário**\_\_\_ presente na pasta *Aparências*, como indicado abaixo :



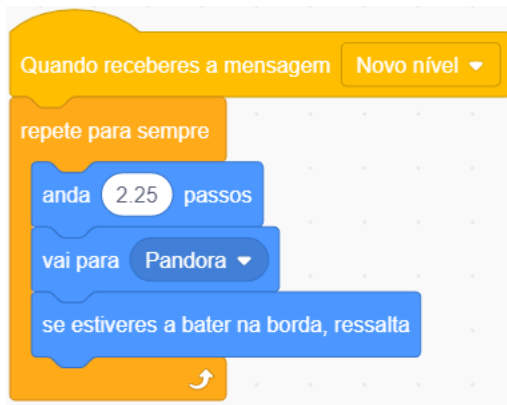
Para certificar-se que o jogo termina e que todos os elementos parem quando a mensagem *Você ganhou!* ou *Você perdeu!* aparece, coloquem o código seguinte no guião Pandora (ou Kraken) :



O usuário tem também de certificar-se que o ecrã certo vai aparecer cada vez que o jogo está jogado ou iniciado ao informar o código seguinte no *Cenário* :



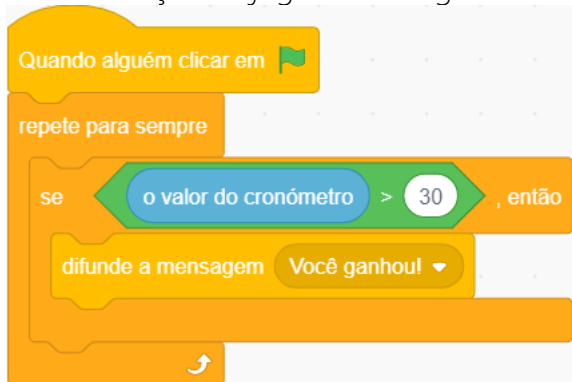
Uma característica do novo nível será o aumento de velocidade do tubarão, tornando o jogo mais difícil para Pandora (usuário). Esta funcionalidade obtém-se ao entrar o código seguinte no guião do Tubarão :



Claro que podem aumentar ainda mais a velocidade modificando os números no bloco *andar* \_\_ *passos*.

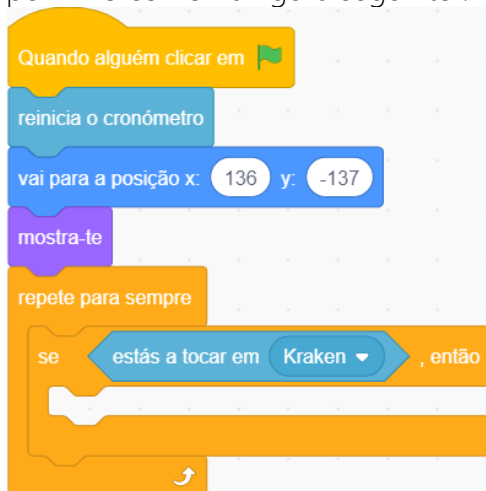
O actor de peixinho deve ficar vivo por um tempo definido escolhido pelo usuário. Se ele conseguir ficar vivo durante este intervalo atribuído , então ele será dado como vencedor.

Para realizar-se entrem o código seguinte no pequeno guião de peixe baseado numa duração de jogo de 30 segundos :



O *Cronómetro* deve voltar ao 0 em cada novo jogo.

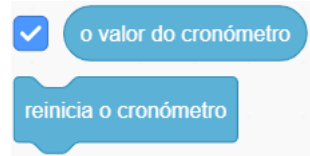
Para realizar-se, coloquem a opção *Voltar a Inicializar o cronómetro* no código do peixinho como na figura seguinte :





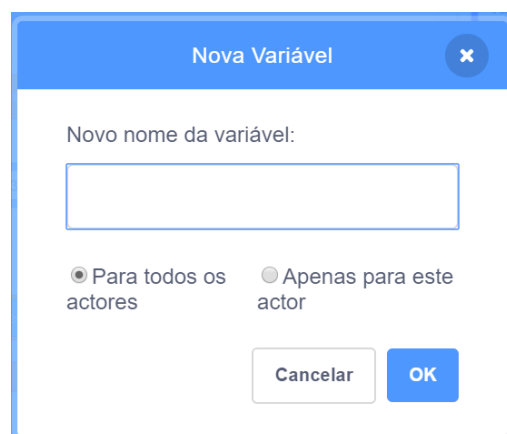
Cliquem na Bandeira verde para verificar que o jogo funcione corretamente. Reparem que uma caixa de indicação do cronômetro aparece na parte superior esquerda do ecrã. **valor do cronómetro 74.414**

Mas, para visualizar o temporizador no palco, o *quadradinho* na esquerda do código cronómetro na pasta Sensores deve ser marcada. Caso não for feito, marquem-na .



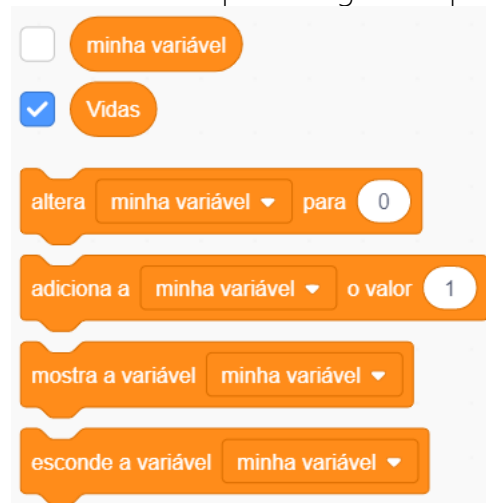
### O uso de variáveis

Para programar Vidas adicionais para o actor do peixinho, vão até a pasta Variáveis e seleccionem a opção *Criar uma variável*. Terão de entrar um nome.



Entrem a palavra *vidas*

O ícone **Vidas** aparece agora na pasta *Variáveis*



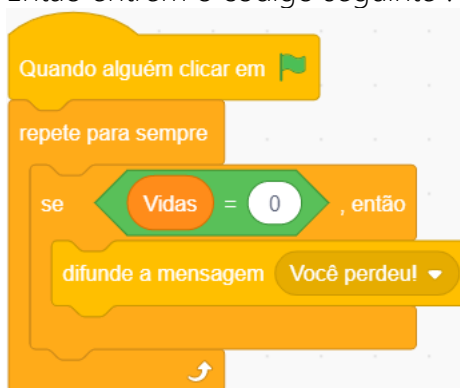
Entrem o número 5 no bloco **Dar vidas em** \_\_ antes de colocá-lo no guião do peixinho como na figura seguinte :



O peixinho agora tem 5 vidas.

Se ele perder todas as suas vidas *antes* de atingir 30 segundos, o jogo termina-se.

Então entrem o código seguinte :



Que está constituído com os blocos nas pastas *Controlo*, *Variáveis* e *Operadores*.

O peixinho perde uma vida cada vez que ele é atingido pelo tubarão.



Obtém-se este resultado inserindo na secção blocos *tecla* do guião do tubarão.



Além disso, retirem o bloco do guião e apaguem-lo.

Com efeito, pelo contrário da antiga versão de *Ataque de tubarão*, este jogo não termina na primeira vez que o peixinho é atingido pelo tubarão pois ele dispõe de 5 vidas a partir de agora.

```

Quando alguém clicar em [bandeira]
  repete para sempre
    se estás a tocar em Pandora, então
      altera minha variável para 1
      altera o teu efeito olho de peixe para 100
      muda o teu traje para shark2-b
      diz Nham! durante 2 s
      muda o teu traje para shark2-a
  
```

Para dar uma oportunidade ao peixinho de tentar de escapar dos dentes do tubarão, temos de adicionar uma faixa de código que lhe permitirá aparecer de forma aleatória em diferentes lugares cada vez que ele for atingido pelo tubarão.

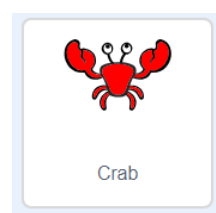
```

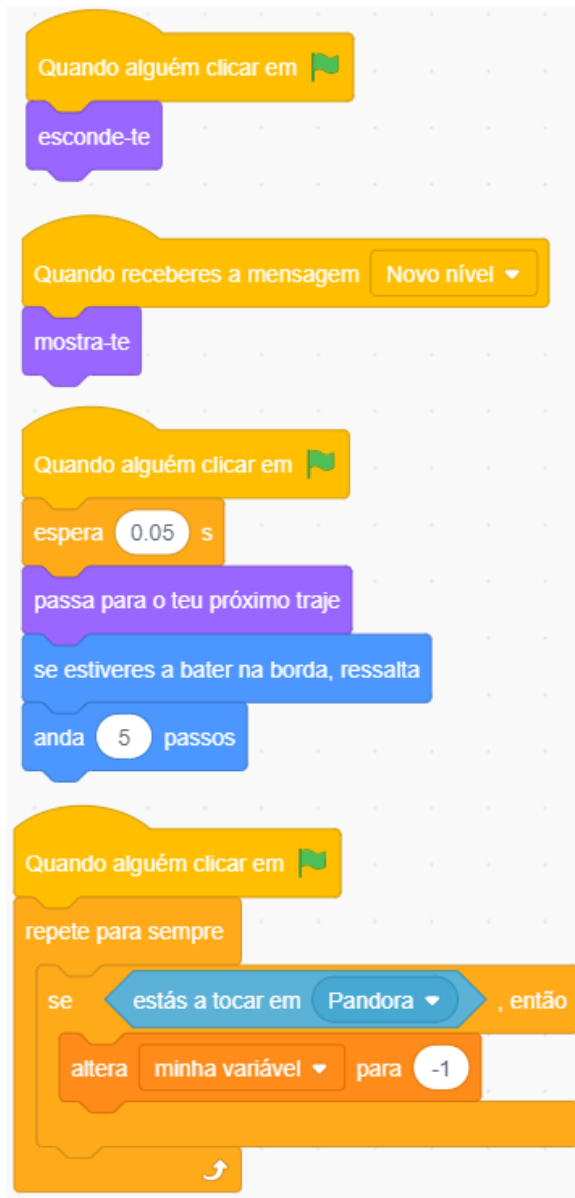
Quando alguém clicar em [bandeira]
  vai para a posição x: 10 y: -118
  reinicia o cronómetro
  mostra-te
  altera minha variável para 1
  repete para sempre
    se estás a tocar em Kraken, então
      esconde-te
      altera o teu x para um valor ao acaso entre -200 e 200
      altera o teu y para um valor ao acaso entre -200 e 200
  
```

Quanto mais ampla for a faixa no bloco número aleatório( verde), maior será a zona de reaparecimento do peixinho (quer dizer *mostrar*) depois de cada ataque do tubarão realizado.

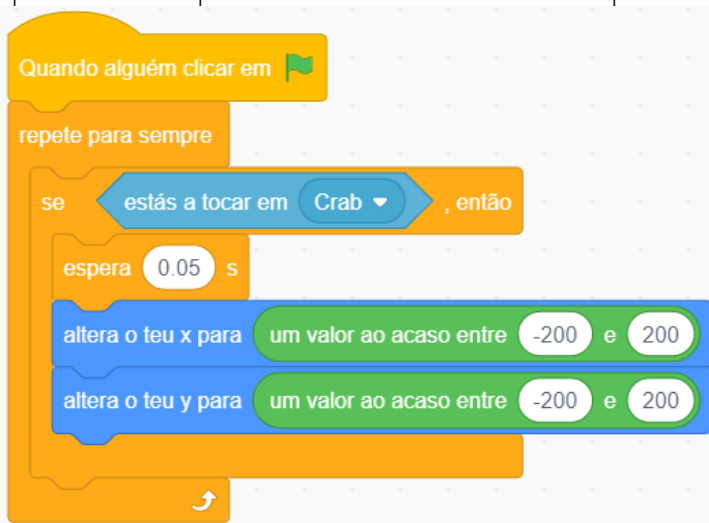
Para acrescentar ainda mais dificuldade ao jogo, criem outro actor que apareça quando o segundo nível for alcançado. Este actor será codificado de modo que ele retire também uma vida se ele atingir o peixinho.

No exemplo abaixo, usamos um actor de *caranguejo*.





Voltemos ao guião do peixinho (Pandora) onde vamos acrescentar as instruções que levarão à perda de uma vida cada vez que ele for atingido pelo caranguejo.



## Exercício

### Perseguir os dinossauros

Peçam aos participantes que criem o jogo do dinossauro perseguidor de outro dinossauro com o uso de *Vidas* e *Níveis*. Portanto, será um actor que irá de uma floresta para um deserto logo irá para um palco d' água, de campo ou de montanha. A partir do segundo palco, peçam-lhes que insiram um segundo dinossauro( ou outra criatura exótica) que começará a perseguir o personagem central.

## Lição 23 – Jogo de ténis solitário

Esta lição apresenta um jogo simples inspirado por um dos grandes jogos clássicos conhecidos como *Pong*. Apresentará aos alunos outro comando de Sensores baseado no *atingir cores*. Os alunos aprenderão também como mudar a direcção dum Actor com uma combinação de comandos de *Movimento* e de *Operadores*.

Uma vez que abrirem Scratch, apaguem o actor do gato.

### O guião Raquete

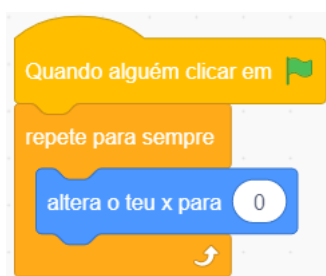


Vão até o ícone novo actor e desenhem uma raquete preta usando a ferramenta de desenho *linha* assim como um *tamanho de pincel* adequado.



Renomeiem o actor « *raquete* ».

Vão ao guião para a raquete.  
Insiram as faixas de código seguintes :



O bloco **Botar x em 0** está na pasta *Movimento*.

Seleccionem **o x da posição do rato** a partir da pasta *Sensores* e arrastem-no no **0 de**.

Assim garante-se que o cursor do rato se deslocará sempre ou na esquerda ou na direita no ecrã no eixo X.

## Os golos – Preparar a quadra

Para criar uma linha de golo, vão ao ícone *Palco* e escolhem a opção *Cenário*. E seleccionem a ferramenta de desenho *Linha*, um grande *tamanho* de *Pincel* e a cor **VERMELHA** na *paleta*.

Arrastem o cursor na parte inferior do ecrã e desenhem no comprimento uma linha grossa vermelha da direita para a esquerda.

Coloquem a raquete um pouco acima da linha de golo VERMELHA .

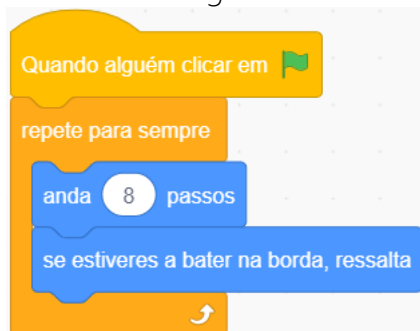
## O guião da bola



Vão até a coluna de figuras dos actores e seleccionem um actor de bola na galeria. Podem também desenhar sua própria bola. Neste caso, não escolham a cor vermelha pois já usamos esta cor na linha de golo.

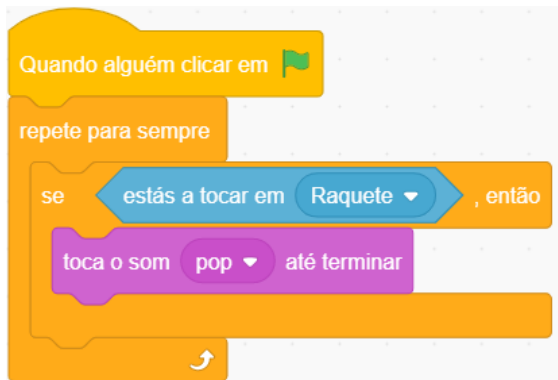
Recomenda-se contudo aos alunos o uso dum actor que já existe no separador para este projecto.

Entremos agora um guião de movimento para a bola

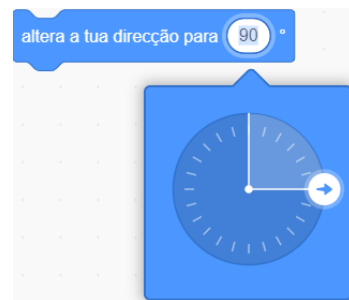


Acompanhado dum bloco de comando separado que permitirá à bola ressaltar numa direcção geralmente ascendente com um efeito sonoro lindo se atingir a raquete.

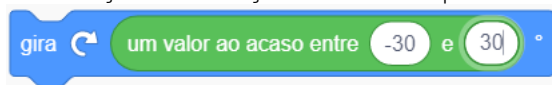
Realiza-se entrando primeiro o guião abaixo mostrado:



No entanto, para que a bola ressalte para cima melhor que para baixo, quando atingir a raquete, temos de usar o bloco *altera a tua direcção para*, onde o movimento para baixo (quer dizer 180) cancela-se ao removê-lo da sua trajectória atual (direcção), ao usar o código adequado na pasta Operadores.



A variação de direcção no seu movimento ascendente obtém-se pelo uso duma selecção de *número aleatório* que, com a faixa de parâmetros entrar, garante uma mudança de direcção cada vez que a bola atinge a raquete.



Seu empurrão para cima é facilitado pelo aumento do número de passos que dá depois de ter atingido a raquete.



A seqüência de código final deste comando deveria ser a seguinte :

A disposição dos blocos de codificação, ilustrada no expmlo abaixo, é muito importante e fundamental no desenrolar certo do jogo.

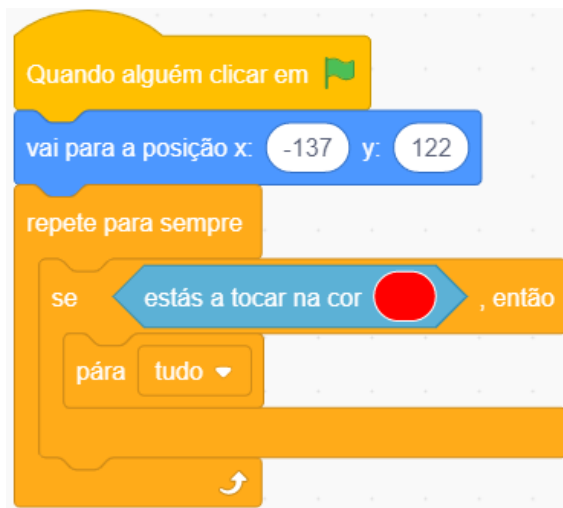
Por exemplo se



Aparece *depois* do bloco, andar **12 passos**, o actor de bola continuará indo para baixo ou de lado melhor do que para cima.

Para garantir que o jogo termine corretamente quando a linha vermelha for alcançada, entrem os comandos seguintes :





**OBSERVAÇÃO:** para que o bloco de código acima mostrado funcione corretamente, a cor vermelha escolhida como **matiz atingida** deveria ter o mesmo valor de cor (*matiz, saturação e luminosidade*) que o do valor de cor vermelha para a linha vermelha.

## Lição 24 – Jogos de aventura: o labirinto incrível !

Os jogos de **aventura** estão baseados num personagem principal que vai fazer uma viagem por numerosas paisagens onde ele /ela enfrenta obstáculos ou perigos que têm de serem superados ou contornados para chegar ao seu destino final e exigir um tesouro ou uma recompensa.

O labirinto é uma forma popular derivada deste tipo de jogo .

Esta lição vem reforçar vários comandos importantes aprendidos nos capítulos anteriores, tais como o **envio de mensagens** e o **atingir a matiz**

### Bitmaps e vectoriais – uma visão rápida

Os Bitmaps e os Vectoriais são diferentes tipos de gráficos em duas dimensões que são usados no Editor de desenho de Scratch.

O que os torna diferentes é a forma que têm de armazenar suas informações.

 Converter para Bitmap

As imagens Bitmap armazenam as informações de cor. Elas armazenam as cores de cada pixel que as constituem.

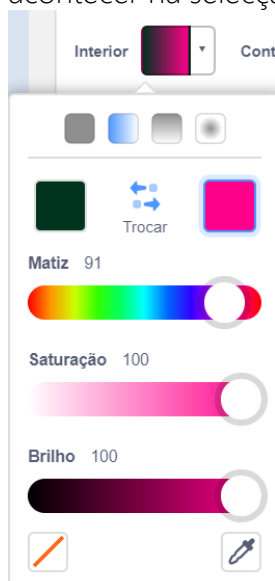
As imagens bitmap armazenam informações de cor de cada pixel que compõe a imagem abaixo mostrada.

 Converter para Vectorial

As imagens vectoriais armazenam as fórmulas matemáticas que desenharam as linhas e as curvas.

### Seleção das cores no Editor de desenho

No momento da selecção da cores no editor de desenho, é fundamental que os utilizadores anotem os números que aparecem em *Matiz*, *Saturação* e *Luminosidade* (ver figura abaixo) . No caso contrário, alguns problemas poderiam acontecer na selecção das cores nos blocos de Sensores para a criação dum guião.



## Criar níveis

Selecione primeiro o ícone Palco, clique em Editar e preencham o cenário de preto.

Copiem este cenário preto três vezes .

Num dos cenários usem a ferramenta TEXTO para escrever **Você ganhou !**

Num segundo cenário, usem a ferramenta TEXTO para escrever **Você perdeu !**

Vão até um dos últimos cenários (Nível básico)

Usem as ferramentas Rectângulo, cor branca e o ícone de contorno para desenhar um determinado número de caixas diferentes localizações do ecrã. No canto superior deslizem um rectângulo amarelo.

Vão até o último cenário e renomeiem-no **Nível avançado**


Coloram o espaço principal de **Verde**.

Usam as ferramentas Rectângulo, cor branca e o ícone completo para desenhar duas nuvens e duas formas de castelo.

## Os actores

Começam seleccionando um actor de personagem principal com um movimento controlado pelas quatro teclas de seta.




Vão até o ícone do novo actor. . Desenhem uma bola colorida usando a ferramenta *círculo*.

Usem o ícone pincel para desenhar um determinado número de linhas irregulares procedendo da zona do círculo.



Dêem o nome de Herói para este actor.

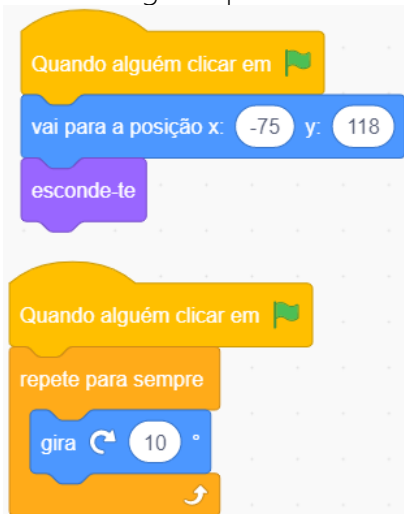


De novo, usem  para desenhar outra bola colorida (VERMELHA) usando a ferramenta de desenho *círculo*. Desenhem depois com o pincel algumas linhas desiguais AZUIS procedendo do círculo.



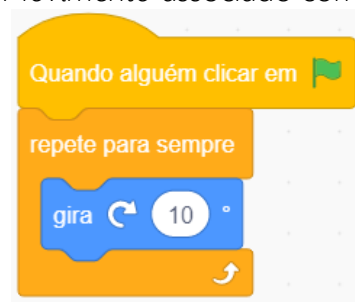
Podem também seleccionar os actores adequados na galeria. Será mais fácil e evitará problemas ao docente pois os alunos não irão esquecer os números específicos da selecção de cores (Matiz, Saturação e Luminosidade) aplicados aos actores criados por eles.

Vão até o guião para criar este actor e entrem os comandos seguintes :



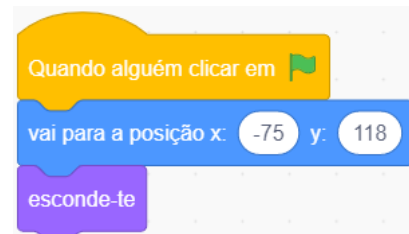
Este actor representa um obstáculo para o actor Herói que, se atingi-lo ,o levará a perder o jogo.

Por essa razão, para aumentar a dificuldade de Herói, teremos o segundo actor numa rotação constante, obtido pelo uso dum bloco *Girar de graus* na pasta *Movimento* associado com um bloco *Repetir para sempre*.

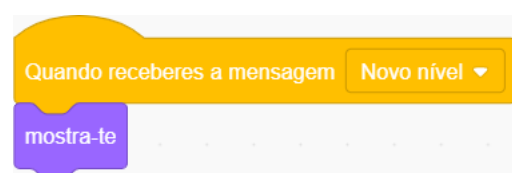


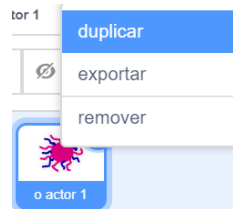
Além disso,este actor aparecerá *somente* no nível avançado.

Por isso foi usado ESCONDER no código de abertura.



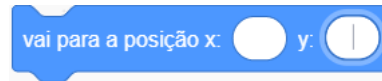
E entrando o bloco seguinte:





Duplicuem este actor seis vezes.

É claro que, a posição de cada um destes actores duplicados deve ser diferente e deve ser situada de tal forma que proporcione um efeito de obstáculo máximo ao actor Herói. Então mudem a em cada bloco individual

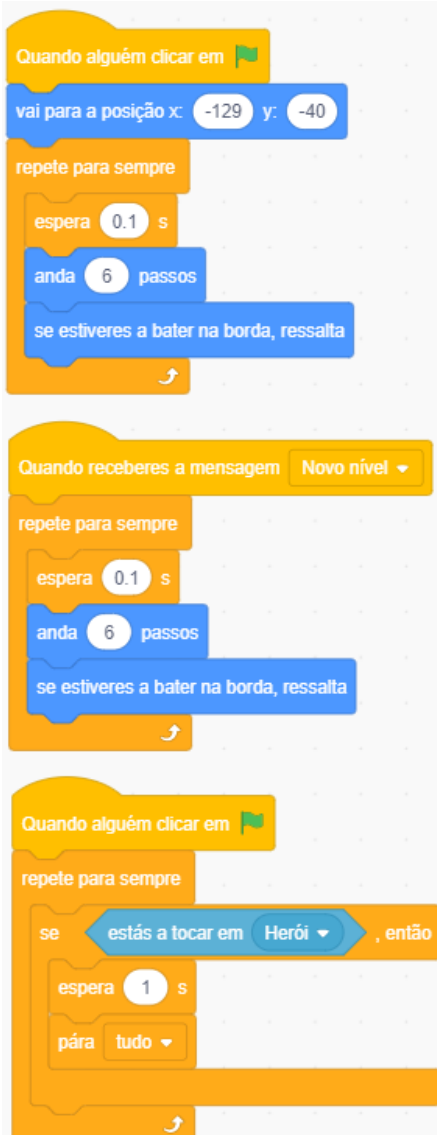


Agora vamos introduzir outro actor obstáculo que será programado para deslocar-se pelo ecrã, para aumentar a velocidade quando o Nível Avançado for alcançado e que ele for atingido por Herói, o jogo vai **terminar** e a mensagem Você perdeu



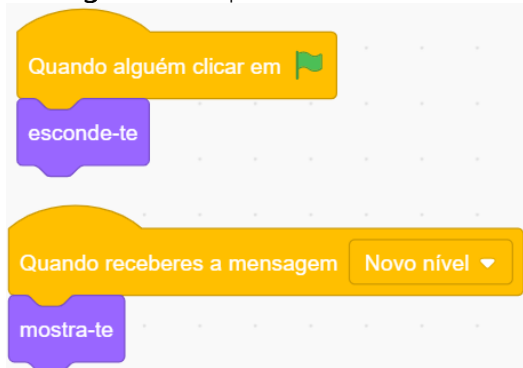
vai aparecer no ecrã preto.

Isso realiza-se usando as três faixas de códigos seguintes para compor seu guião :

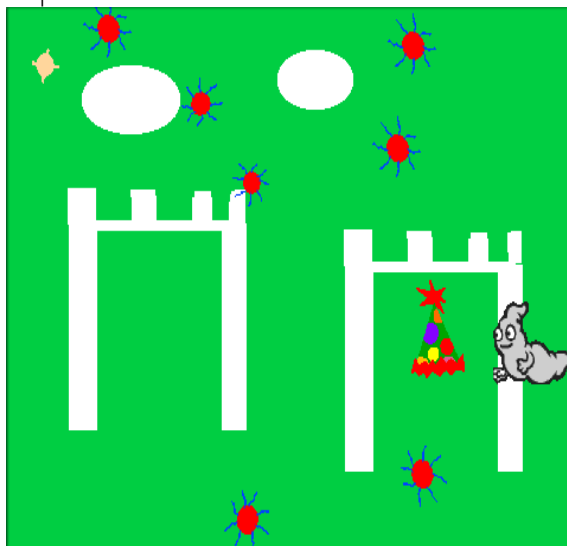


Para aumentar o nível de dificuldade para o actor Herói, o actor Fantasma pode ser duplicado várias vezes, com coordenadas X e Y obviamente distintas em cada versão.

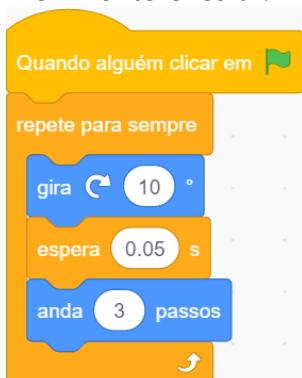
A adição do actor final irá servir de PRÊMIO. Se o actor Herói alcançar a mensagem **Você ganhou !** aparecerá inscrito no ecrã e o jogo vai terminar.



O posicionamento final dos actores no Nível avançado tem de ser similar a :



A intensidade da dificuldade para Herói ganhar o PRÊMIO pode ser aumentada, por exemplo, ao adicionar dois blocos de código ao guião do actor de círculo vermelho posicionado na frente de PRÊMIO, o que lhe permitirá deslocar-se em movimento circular.

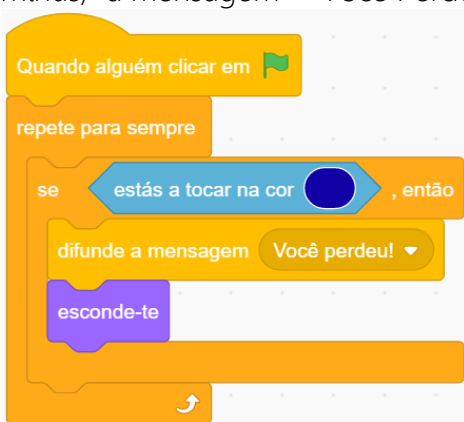


O guião para Herói teria de incluir :

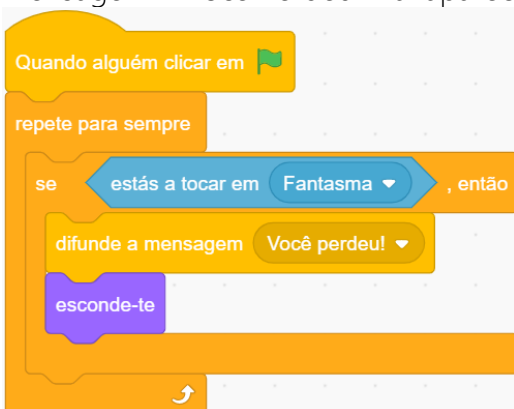
Um pacote de instruções indicando que no caso de alcançar as paredes *brancas* do labirinto, Herói desaparecerá.



Um pacote de instruções indicando que em caso de ele atingir os *picos azuis das minas*, a mensagem « Você Perdeu » vai aparecer e Herói desaparecerá.



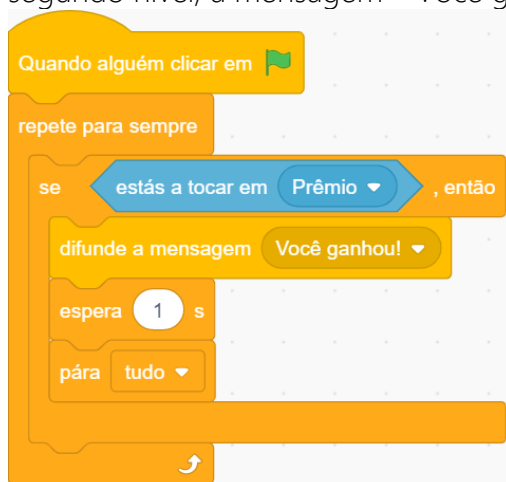
Um pacote de instruções indicando que no caso de ele atingir o actor *Fantasma*, a mensagem « Você Perdeu » vai aparecer e Herói desaparecerá.



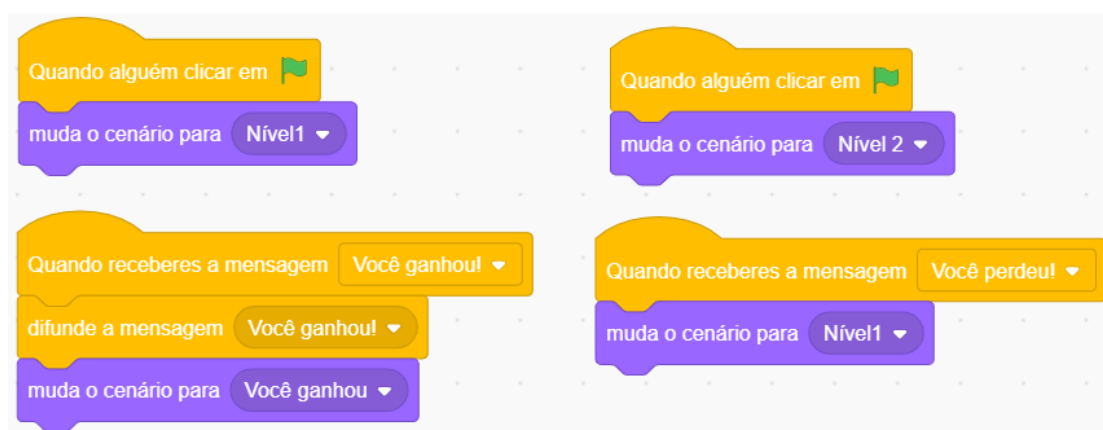
Um pacote de instruções indicando que em caso de ele atingir a caixa *amarela*, o jogo passará ao nível seguinte.



Um pacote de instruções vai indicar que no caso de ele conseguir o prêmio no segundo nível, a mensagem « Você ganhou » vai aparecer e o jogo vai terminar.



Por fim, o código do palco deve ser escrito para responder (via a integração dum bloco **Quando eu recebo** ) aos « **Enviar para** » incluídos no actor Herói:



### Exercício :

**Digam « Cheese » !** Peçam aos alunos elaborarem um projecto Labirinto em que um rato deve evitar de cair nas armadilhas contra ratos e evitar de ser capturado por gatos para atingir a linha de saída e curtir uma bela e grande fatia de queijo .



## Lição 25 – Caçador de demônios

### Caçador de demônios



#### Condições de realização do jogo- Resumo do plano de codificação

- O « Actor de bem » desloca-se com as teclas de seta mas tem de avançar sempre
- O demônio desloca-se de forma aleatório para que o actor de bem não saiba onde ele vai aparecer
- O actor de bem começa com 5 vidas
- Se o demônio atingir o actor de bem , este vai perder uma vida
- Se as vidas do actor de bem têm o valor equivalente a zero, então o jogo termina e um cenário « Game over » vai aparecer.
- Quando o jogo volta a iniciar, o cenário volta a inicializar-se e as vidas também voltam a se inicializar em 5.

**Blocos de comandos chaves:** « Repetir para sempre \_\_\_\_ », « Quando eu recebo », « Difundir para » (pasta Controlo), « Dizer » (Aparências), « Atinge-o (um actor) » (Sensores), « Número aleatório entre \_\_\_ e \_\_\_ », « Criar uma Variável », « Criar uma Lista » (Dado), « \_\_\_ou \_\_\_ » (Operadores), « Girar de \_\_\_ graus », « Ressaltar se a borda for alcançada » (Movimento), « Tocar o som \_\_\_\_ » (Som).

A opção *Criar uma Lista* na pasta *Dado* permite ao programador compilar, por exemplo, uma lista de comentários ou de palavras que um actor diria quando algo específico acontece, como ser atingido por outro actor durante um jogo.

O jogo precisa de dois actores e de duas versões de cenário. Este cenário teria um ecrã visualizando o texto **Game Over**.

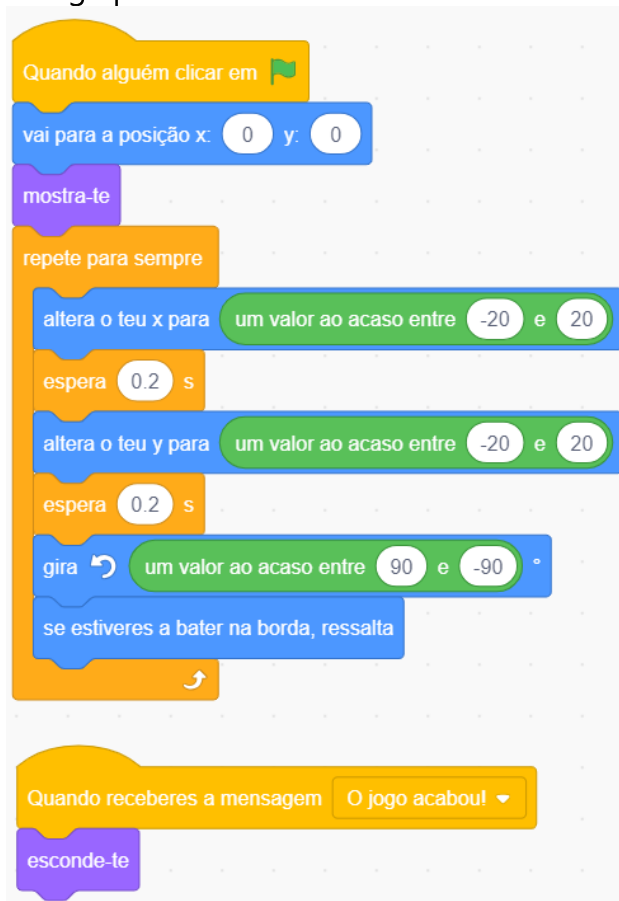
*Melhorias* : Após ter executado as instruções acima citadas, peçam aos alunos efetuarem alguns ajustes que eles achariam convenientes para melhorar o jogo.

### Código (Instruções) Para Gobo (o 'actor de bem')

The image displays four distinct Scratch code blocks for Gobo:

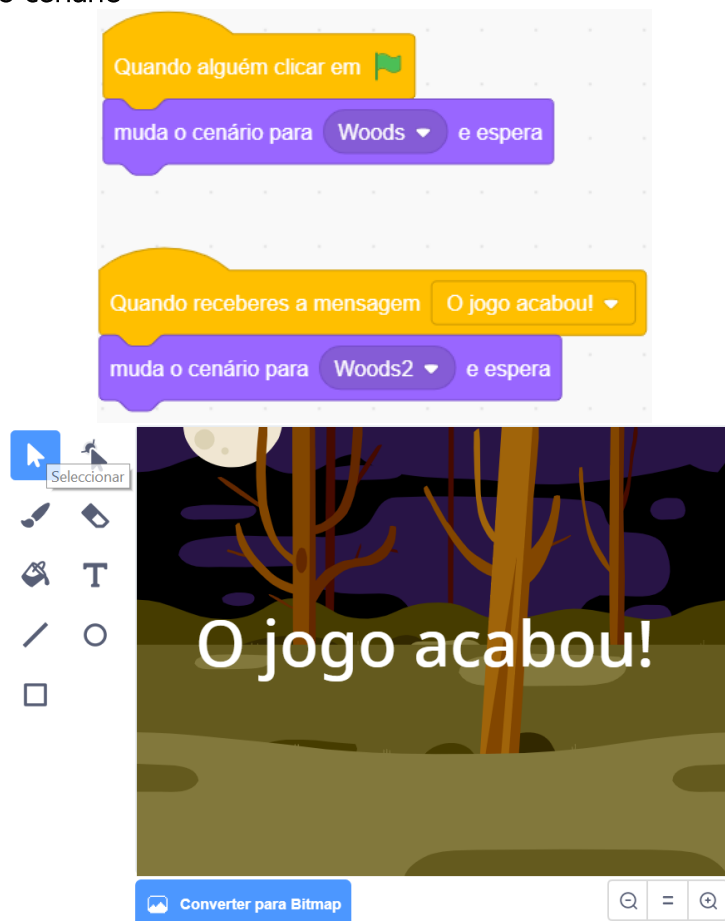
- Block 1:** Starts with a 'Quando alguém clicar em' (When clicked) event. It moves Gobo to x: -186, y: -147, shows it, and sets 'Vidas' (Lives) to 5. A 'repete para sempre' (Repeat forever) loop contains a 'se estás a tocar em Demônio' (If touching Demon) condition. Inside the loop, it changes the 'remoinho' (Spin) effect to 120, says '2' for 2 seconds, changes the 'remoinho' effect to 0, and subtracts 1 from 'Vidas'.
- Block 2:** Starts with a 'Quando alguém clicar em' (When clicked) event. A 'repete para sempre' (Repeat forever) loop contains a 'se Vidas = 0' (If Lives = 0) condition. Inside the loop, it broadcasts the message 'O jogo acabou!' (The game is over!).
- Block 3:** A 'Quando alguém pressionar a tecla seta para' (When key pressed) event for the 'seta para baixo' (down arrow) key, which adds 10 to 'x'. A second 'Quando alguém pressionar a tecla seta para baixo' (When key pressed) event for the 'seta para baixo' key, which adds -10 to 'x'.
- Block 4:** A 'Quando alguém pressionar a tecla seta para' (When key pressed) event for the 'seta para' (right arrow) key, which adds 10 to 'x'. A second 'Quando alguém pressionar a tecla seta para a esquerda' (When key pressed) event for the 'seta para a esquerda' (left arrow) key, which adds -10 to 'x'.

## Código para o demônio



```
Quando alguém clicar em [bandeira verde]
vai para a posição x: 0 y: 0
mostra-te
repete para sempre
  altera o teu x para um valor ao acaso entre -20 e 20
  espera 0.2 s
  altera o teu y para um valor ao acaso entre -20 e 20
  espera 0.2 s
  gira um valor ao acaso entre 90 e -90 °
  se estiveres a bater na borda, ressalta
Quando receberes a mensagem O jogo acabou!
esconde-te
```

## Código para o cenário



```
Quando alguém clicar em [bandeira verde]
muda o cenário para Woods e espera

Quando receberes a mensagem O jogo acabou!
muda o cenário para Woods2 e espera
```

The image shows a Scratch interface with a scene preview. The scene is a dark forest at night with a full moon and the text "O jogo acabou!" (The game is over!) in white. A toolbar on the left includes a selection tool labeled "Seleccionar". At the bottom, there is a "Converter para Bitmap" button and zoom controls.

## Exercício

Peçam aos alunos criarem seu próprio jogo para a próxima aula, com base no que já aprenderam nesta lição assim como em outras lições, e com o uso de actores desenhados por eles mesmos.

Incentivem cada pessoa ou cada grupo a explicar e a apresentar demonstrações das suas próprias criações de projectos para toda a turma.

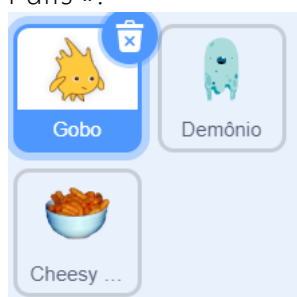
## Lição 26 - Extensão do jogo Caçador de demônios

### Condições de realização do jogo - resumo do plano de codificação

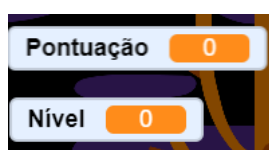
Nesta lição, os alunos vão prolongar o jogo Caçador de demônios adicionando tesouros a serem colecionados pelo actor de bem (Gobo), mantendo uma pontuação e modificando os níveis quando a pontuação atingir determinados valores. Um jogador perderá o jogo se o número de vidas alcançar zero e ganhará se o nível 3 for alcançado.



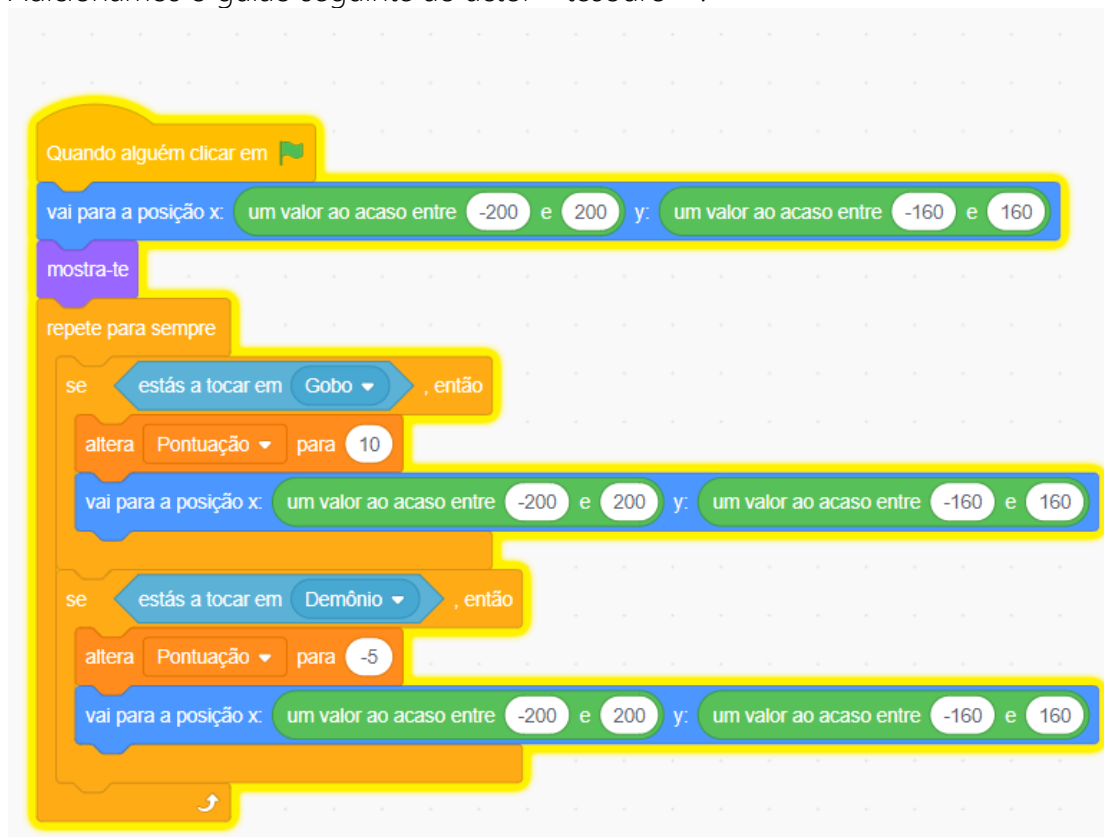
Adicionamos primeiro um novo actor para o tesouro( aqui uma tigela de « Cheesy Puffs »).



E acrescentamos duas novas variáveis, uma chamada «Pontuação» e a outra «Nível».

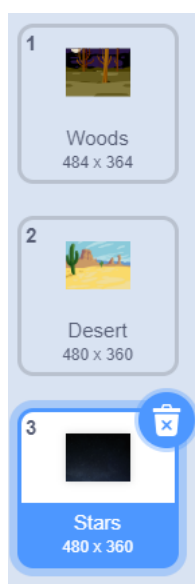


Adicionamos o guião seguinte ao actor « tesouro » :



Este guião desloca o actor tesouro até uma parte aleatória do ecrã quando o jogo começa. Cada vez que Gobo atinge o tesouro, a pontuação aumenta de 10 pontos e o tesouro está deslocado até uma nova posição aleatória. Se o Demônio atingir o tesouro então a pontuação reduz-se de 5 pontos e o tesouro está deslocado para uma nova posição aleatória.

Agora, vamos apagar o cenário « woods1 » originário e que leva a mensagem "Game Over ".Vamos adicionar 2 cenários ao palco. Quando o nível mudar , mudaremos o cenário.



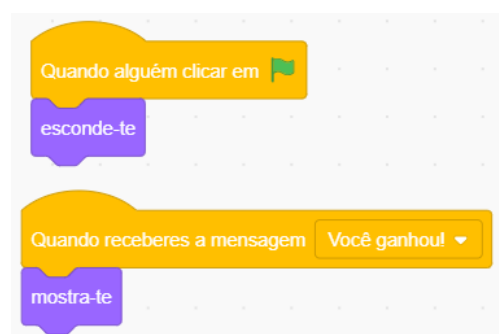
Adicionamos o guião seguinte ao palco para controlar os níveis:



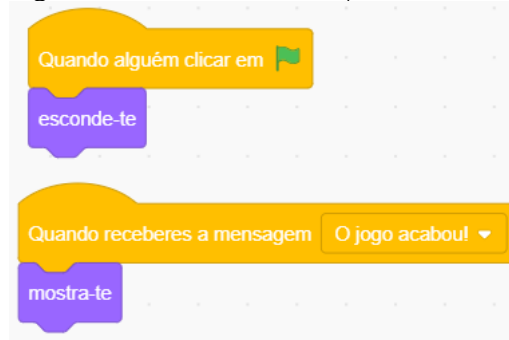
Quando alguém clica na bandeira verde, estabelecemos a pontuação em 0, o nível em 1 e o cenário em « woods » (o primeiro cenário). Adicionamos uma repetição *repetir para sempre* para continuar a verificar a pontuação e para passar ao nível seguinte caso for necessário. Se estamos no nível 1 e que a pontuação estiver além de 50, então passamos ao nível 2 e alternamos no cenário seguinte. Da mesma forma, se estamos no nível 2 e que a pontuação estiver além de 100, passamos ao nível 3 e alternamos no cenário 3. Se estamos no nível 3 e que a pontuação estiver além de 150, então o jogador ganha e difundimos uma mensagem « Você ganhou ».

Agora adicionamos 2 actores para indicar se um jogador ganhou ou se ele perdeu. Usar actores com este fim permite mostrar simplesmente uma mensagem « ganhou » ou « perdeu » no cenário actual :

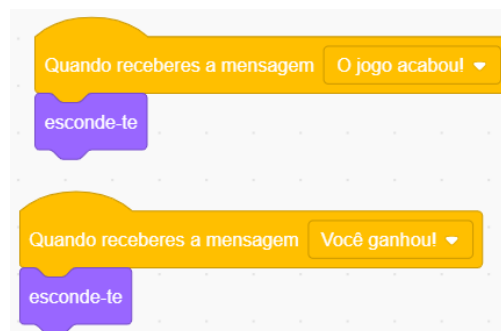
Adicionamos o guião seguinte ao actor « Você ganhou » :



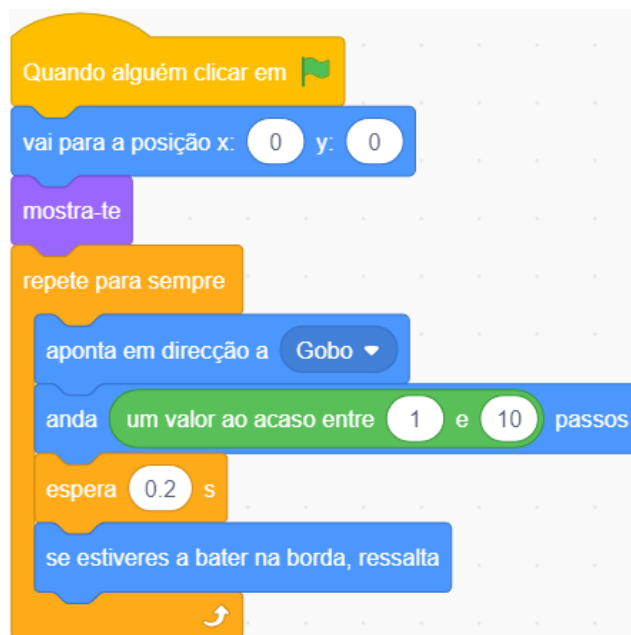
Quando o jogo começa, escondemos o actor « Você ganhou ». Se recebemos a mensagem « Você ganhou » difundida para o palco, então a mostramos. Adicionamos o guião seguinte ao actor « Você perdeu » :



Quando o jogo começa, escondemos o actor « Você perdeu ». Se recebemos a mensagem « Game Over » difundida para o actor de bem, então a mostramos. Adicionamos os guiões seguintes aos actores « actor de bem », « actor mau » e « tesouro » para escondê-los quando o jogo acaba:



Para tornar o jogo um pouco mais difícil, podemos fazer com que o Actor Mau siga ativamente o actor de bem em vez de apenas deslocar-se de forma aleatória. Podemos fazer com que ele se mova mais rapidamente à medida que aumentarem os níveis. Para realizá-lo , modificamos o guião do actor mau da forma seguinte:





Neste guião, direccionamos continuamente o actor mau para o actor de bem e mandamos ele andar por um número aleatório de passos até o actor de bem. O número máximo de passos que o actor mau pode dar é 10 vezes superior ao nível actual. Isso significa que ele tende a acelerar à medida que os níveis aumentam, o que torna o jogo mais difícil em cada nível novo.

### Exercício

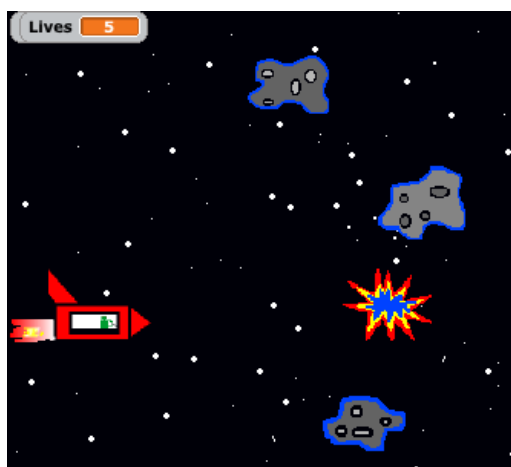
Peçam aos alunos criarem um jogo de aventuras baseado numa lenda de herói da mitologia do seu próprio país/ da sua cultura, que deve combater demónios e monstros nas montanhas, pelos mares e nas cavernas com o fim de ganhar um tesouro.

## Lição 27 – Jogo de atirar: os asteróides

Os jogos de **atirar** estão baseados num ambiente em que o personagem principal ganha pontos para atingir outro personagem ou um conjunto de personagens. Este tipo de jogo é geralmente bastante difícil e pode ele pode testar a velocidade e o tempo de reacção do jogador. Várias vezes, o personagem-jogador, caso ele sobreviver aos obstáculos, não perde vida e reivindica uma determinada pontuação, ele pode logo avançar para outro nível ou outra missão.

### Asteróides

Nesta versão simplificada do jogo clássico Asteróides, os alunos descobrirão pela primeira vez a maneira de criar um actor que imite os movimentos de acção rápidos dum laser( ou de uma bola).



#### Condições de realização do jogo - Resumo do plano de codificação

A astronave entra numa área de asteróides .O piloto deve destruir as pedras astrais para salvar a astronave.

A astronave pode sobreviver somente três vezes antes de ser destruída.

Mas por cada tentativa direta, o piloto ganha um ponto.

Primeiro, seleccionem o cenário « Stars » a partir da pasta *Espaço*.

### O actor astronave

Desenhem (ou seleccionem a partir da galeria dos actores) uma astronave com três Trajes, com um que representa uma explosão.

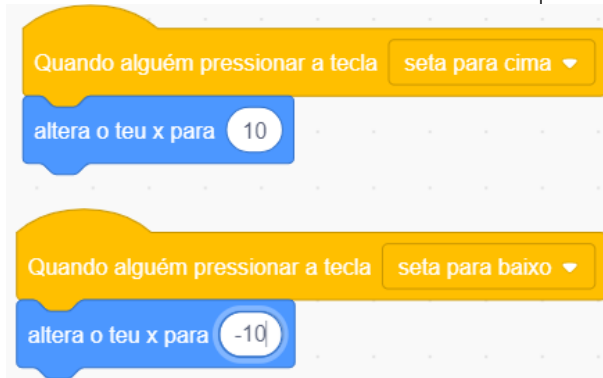
Uma vez que forem codificados os dois primeiros Trajes, eles darão a impressão que a astronave se desloca rapidamente no espaço por causa da pouca diferença de tamanho e de forma da astronave, especialmente por causa da largura ou do comprimento das chamas do booster.



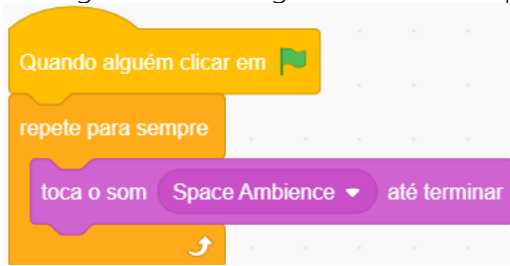
Configurem as *Vidas* pela ajuda da opção *Minha variável* da pasta *Variáveis*.  
 Seleccionem um número máximo de 5 vidas.  
 Criem um monitor de *Pontuação* com o mesmo procedimento com o zero (0) como número de partida.



Controlem o *movimento da astronave* apenas com as teclas de seta *Alto* e *Baixo*.



Carreguem um código de efeito adequado



Desenham um actor de **asteróide** com dois trajes.



No primeiro traje, o asteróide deveria ser de cor cinza.

Seleccionem os números seguintes ao desenhar :

Cor : 25 ; Saturação : 0 ; Luminosidade : 65.

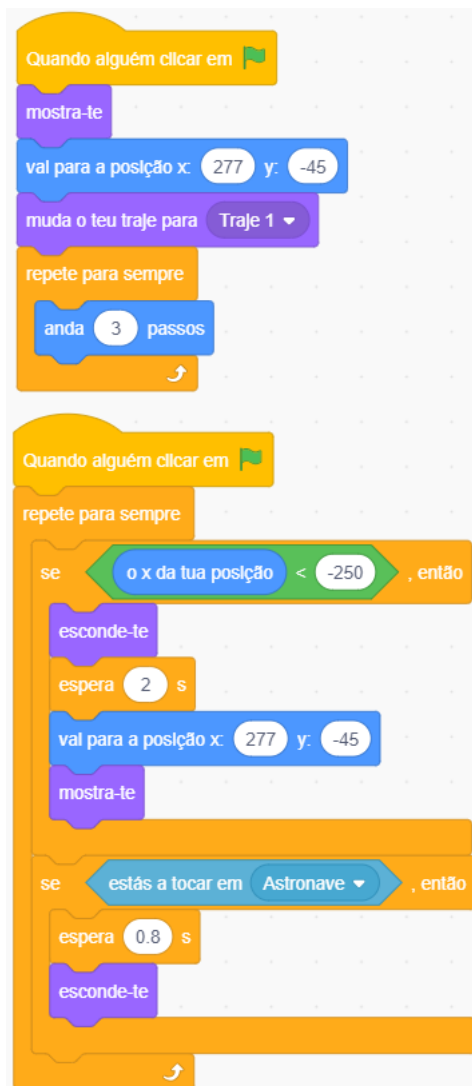
O segundo traje deveria representar uma explosão.

Agora dupliquem o asteróide ao menos quatro vezes.

Cada guião de asteróide deveria ter um eixo X e Y diferente para garantir que os actores não comecem desde o mesmo lugar nem seguem a mesma trajectória.

Além disso, o ambiente será melhor se estas pedras em movimento desaparecerem bem pouco antes de alcançar a borda esquerda do ecrã e de reaparecer um ou dois segundos mais tarde ao sair do lado direito.

Isso realiza-se com o código seguinte (com cada actor de pedra que tenha coordenadas X e Y diferentes) :



Voltemos agora ao guião do Astronave, que temos de programar comandos que vão indicar um embate com um asteróide.

Como existem vários asteróides, seriam necessários muitos códigos para reconhecer cada um dos actores individuais. No entanto, identificamos coletivamente todos os asteróides pelo uso da **cor cinza**.

Portanto o guião de codificação principal do astronave deveria ler-se da forma seguinte :

Por fim, entrem o código que parará o jogo uma vez que todas as vidas do astronave forem perdidas:

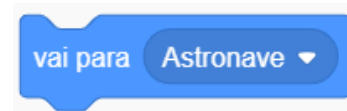
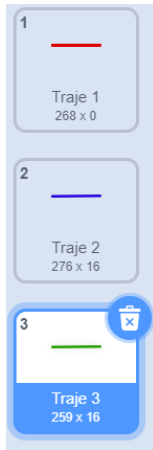
### O Actor Laser

Desenhem um novo actor que está constituído de uma linha curta e grossa.

Copiem este traje seis ou sete vezes .

Dêem uma cor diferente a cada traje .

Isto dará a impressão dum *tiro* quando ele for codificado.



Alinham o actor Laser com a astronave pelo uso de na categoria Movimento.

No entanto, têm de ir ao *Editor de desenho* do traje do actor da astronave para conseguir um alinhamento melhor entre os dois actores.

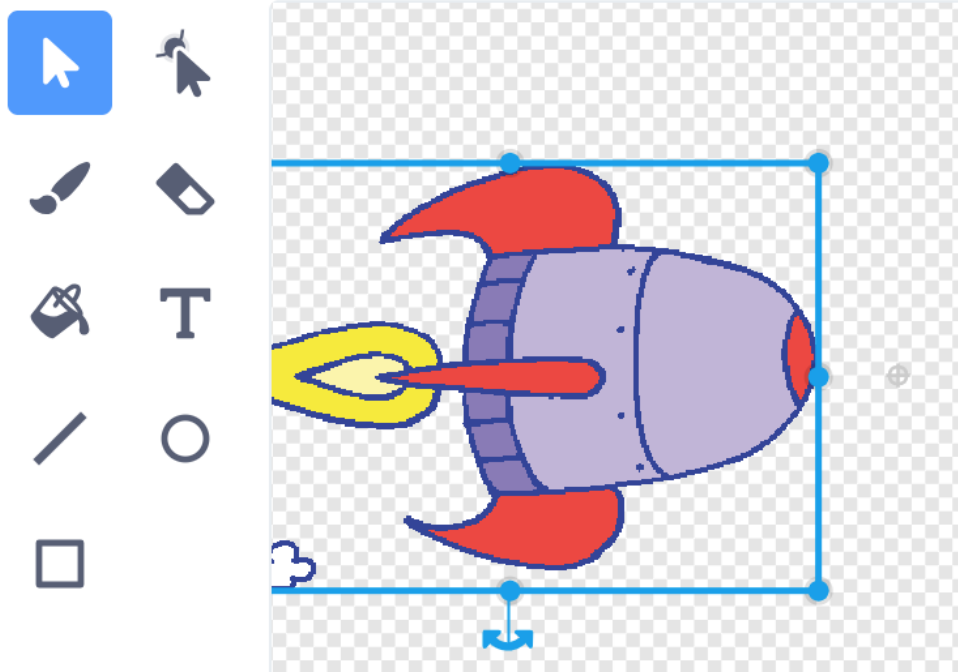
Quando for feito, deslize o actor astronave da parte central do ecrã no *Editor de desenho* com o ícone de posicionamento *Seta* situado no canto superior esquerdo do menu ferramenta no modo *Vectorial*.

Mas em caso de o actor ser composto de vários elementos que podem se deslocar independentemente uns dos outros com esta ferramenta, accessem a ferramenta situada na parte direita de cima do menu e que permite manipulá-lo como se fosse



uma entidade só.

Vão ver uma pequena marca *alvo* . Deslize o actor astronave para esquerda e na frente do alvo.



Comecem o projecto de guião completo e vejam se está a funcionar. Terão de testar várias vezes o deslocamento do actor astronave no editor de desenho até que entendam que o alinhamento entre os dois actores seja satisfatório.

Agora, vamos construir um código que vai dar a impressão visual que uma vez a barra espaço for alcançada, um laser vai ser lançado desde o astronave numa linha de tiro direita com rumo para o asteróide.

```

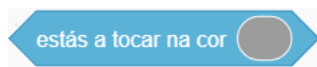
Quando alguém pressionar a tecla espaço
  vai para a posição x: -200 y: -25
  esconde-te
  até que o x da tua posição > 200, repete
    mostra-te
    anda 60 passos
    espera 0.001 s
    passa para o teu próximo traje
  se o x da tua posição > 200, então
    esconde-te
  
```



O código abaixo inclui comandos que gravam as teclas do laser num asteróide:



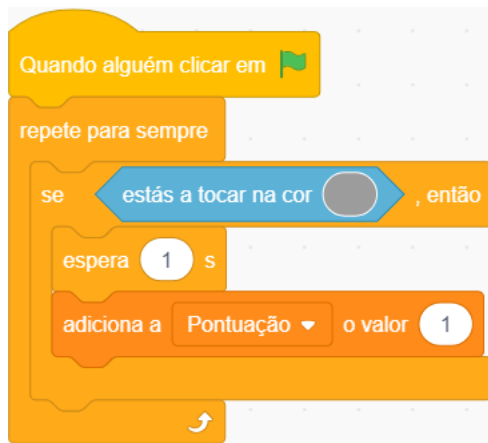
Mas certifiquem-se que no bloco,



Vocês entram os mesmos números na caixa de cor que os colocados no momento da coloração do asteróide. Senão nenhuma pontuação poderá ser gravada ao atingir o asteróide.

Estes números são :

Matiz : 25 ; Saturação : 0 ; Luminosidade : 65.



## Lição 28 – Jogos de dois jogadores

*O trabalho de equipa é uma das forças fundamentais da pedagogia Scratch que recomenda exercícios práticos regulares que envolvam dois estudantes ou mais que realizam o mesmo projeto.*

*Da mesma forma, incentivamos os participantes a criarem jogos que envolvam vários jogadores, pois os jogos de computador modernos tendem a ser feitos em ambiente de redes sociais interativos on-line.*

### Atacante de futebol

Os jogos de tiro são muito populares.

#### Condições de realização do jogo – resumo do plano de codificação

Um jogador, o atacante, tenta de marcar 5 tiros. O outro jogador, o guarda-linha, tenta de defender 5 tiros. O primeiro que realizar seu objetivo ganha o jogo.

O guarda-linha desloca-se com o uso das teclas W e S. O atacante e a bola deslocam-se no eixo Y com as teclas Alto e Baixo. Para dar um tiro, o atacante clica no espaço. O posicionamento destas quatro teclas de movimento permite aos jogadores usarem o teclado ao mesmo tempo.

O programa volta a inicializar-se imediatamente depois dos tiros atingirem seu alvo, fora do alvo ou depois da defesa. Isso continua até que um dos dois jogadores atinja cinco no ecrã de backup/pontuação.

Neste jogo, tem também nosso primeiro encontro com uma sequência de **Perguntas e Respostas**.

Originalmente, este jogo foi programado por Philip, com oito anos de idade.

**Blocos de comandos chaves:** comando de sensores « Fazer uma Pergunta e Resposta », « Matiz atingida », do comando « Se », Variáveis & Operadores, « Quando eu recebo » e « Enviar a todos ».

Anotem que o comando *Solicitar* (pasta de sensores) permite ao utilizador que insira uma resposta. A resposta certa ou falsa e respostas de resultados são controladas pelos comandos inseridos.



Desenhem um jogo adequado a um campo de futebol.

Carreguem actores com papéis de atacante, de guardião, de bola de futebol e de hóspede oficial.

**Remarque :** nos guiões abaixo, todos os posicionamentos X e Y para o atacante, o guardião, a bola de futebol e o hóspede são todas específicas ao projeto de demonstração e não serão compatíveis com nenhum outro projecto. Apenas existem como linhas normativas.

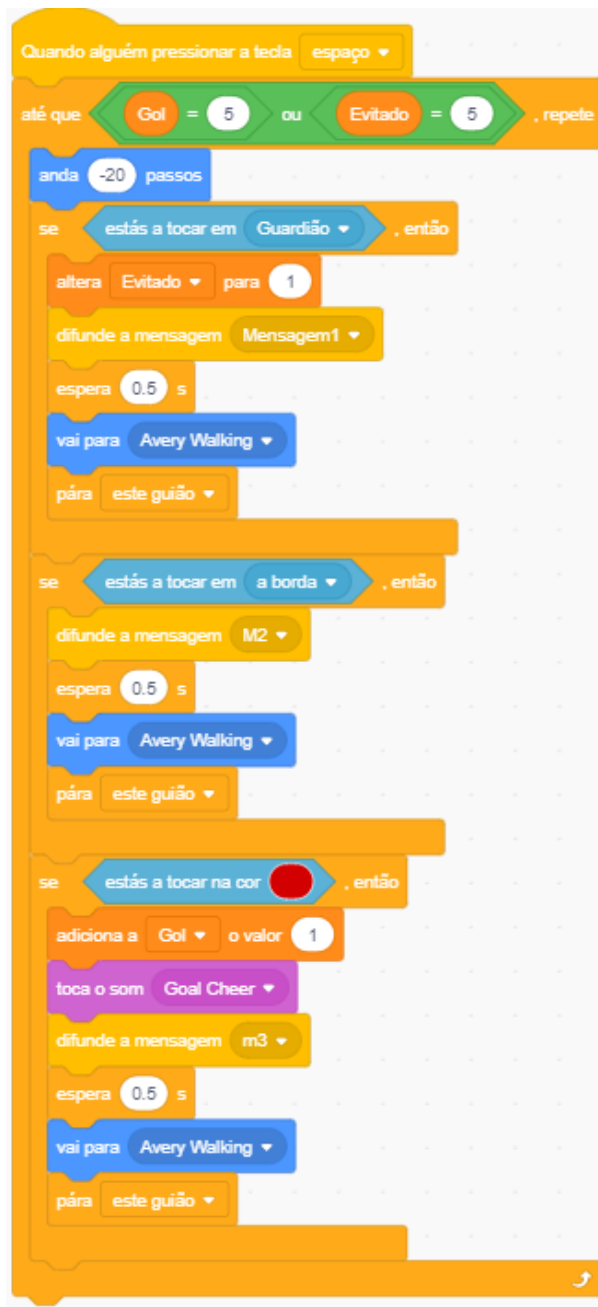
Guião : O actor MC ou Hóspede


```
Quando alguém clicar em [bandeira]
  altera o teu x para -96
  altera o teu y para -114
  mostra-te
  pergunta "Você quer jogar futebol?" e espera pela resposta
  se a resposta = Sim, então
    difunde a mensagem Sim
    diz "Utilize W e S para mover o guardião" durante 2.5 s
    diz "Utilize as setas para cima et para baixo para mover o Atacante" durante 2.5 s
    diz "Pronto?" durante 1 s
  senão,
    diz "O jogo esta cancelado!" durante 2 s
    difunde a mensagem Cancelado!
  esconde-te
```

Guiões : actor bola


```
Quando alguém pressionar a tecla seta para cima
  altera o teu x para 10

Quando alguém pressionar a tecla seta para baixo
  altera o teu x para -10
```



Observação : O  no guião acima mostrado, refere-se a uma referência, a cor das redes de golo para gravar uma pontuação.



No texto abaixo, o bloco  aparece ao lado do atacante.

No entanto, vocês terão de entrar provavelmente no *Editor de desenho* dos trajes do actor de atacante com o fim de colocar a bola aos pés do atacante.

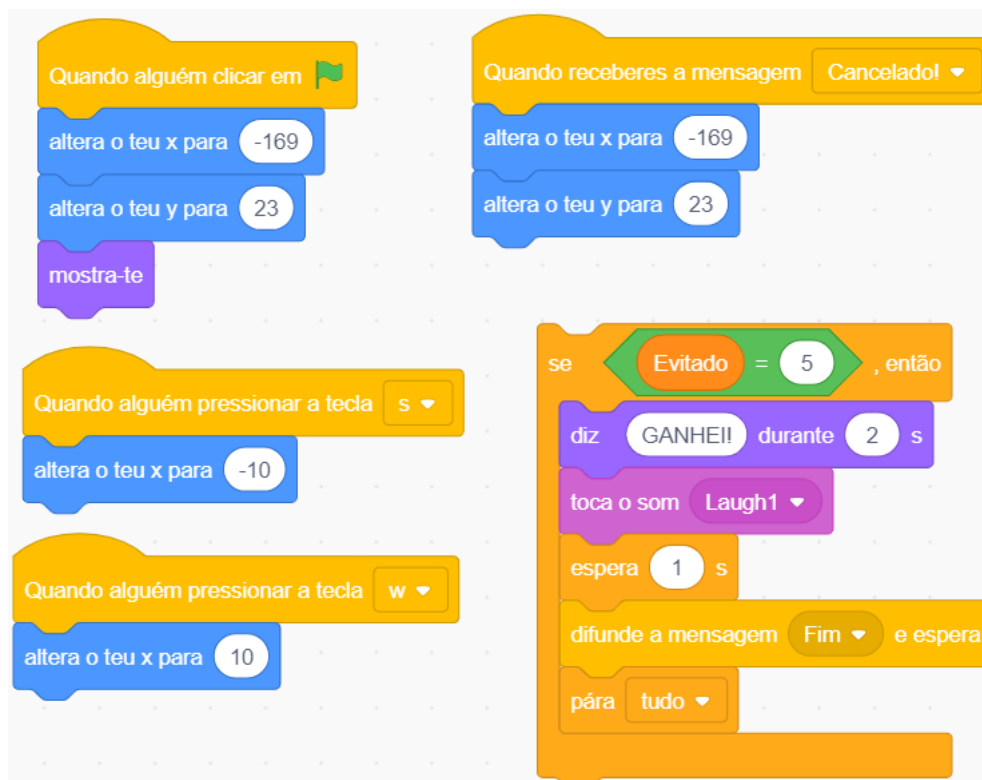
Quando for feito, deslizem o actor do atacante na parte central do ecrã no *Editor de desenho* com o ícone de posicionamento de *seta* situado no canto superior esquerdo do menu Ferramentas no modo *vectorial*. No entanto, se o actor tiver vários elementos que possam se deslocar independentemente uns dos outros com desta ferramenta, accessem a ferramenta situada na parte direita de cima do menu



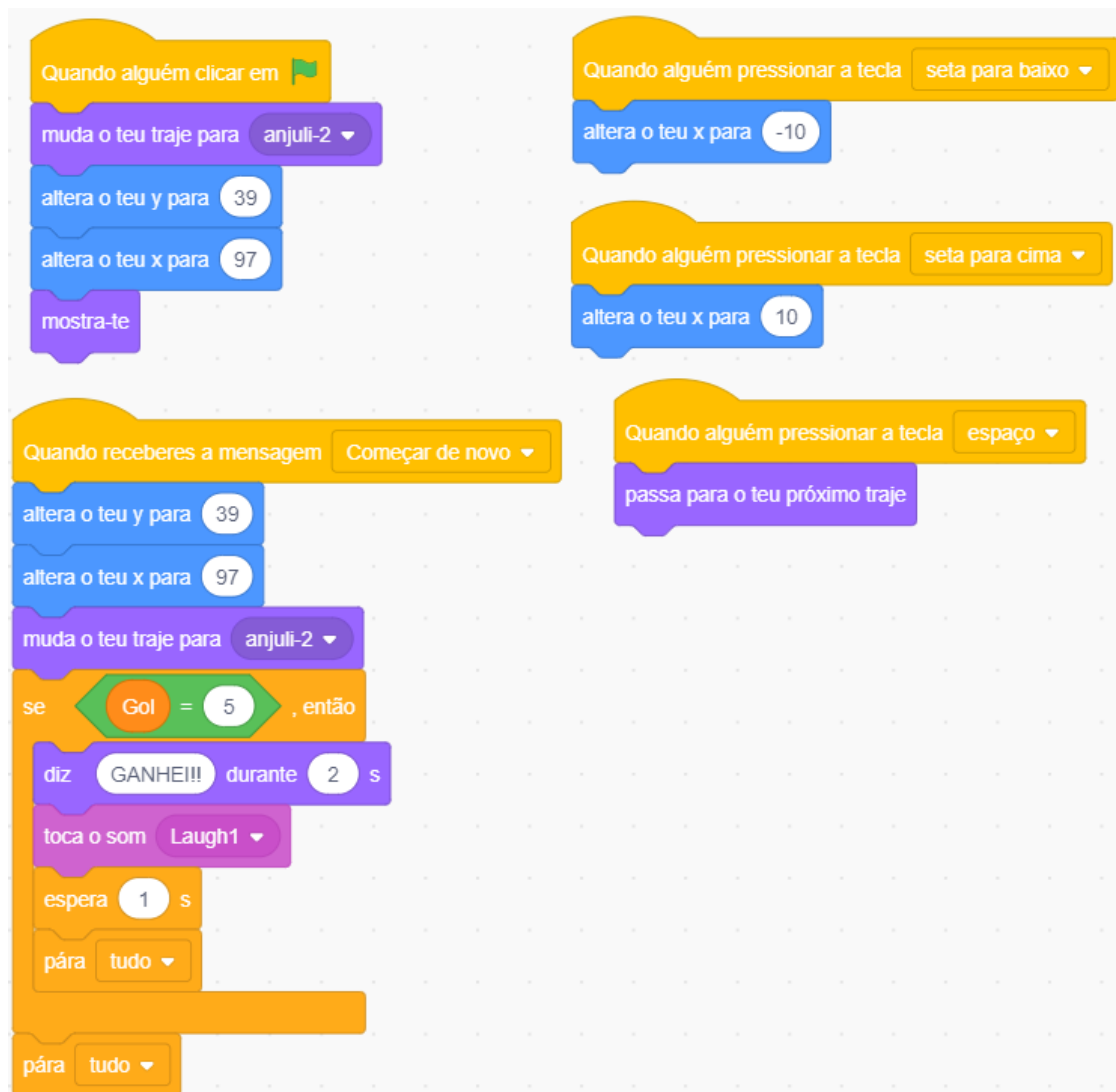
e que permite manipulá-lo como se fosse uma entidade só.

Quando tiverem deslocado o actor, vocês vão ver uma pequena marca de *alvo*. Desloquem o actor atacante até que seus pés fiquem atrás do actor bola. Olhar o palco ajudará a conseguir o alinhamento certo.

### Guiões: actor guardião



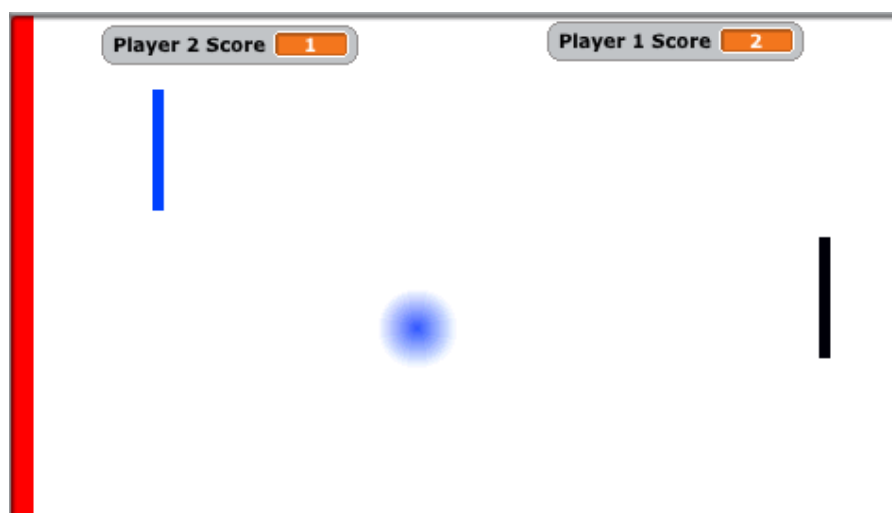
## Guião : actor atacante



## Exercício

Criem um *jogo de hóquei* ou outro jogo desportivo baseado numa estrutura idêntica ou similar ao jogo Atacante de Futebol acima mostrado.

## Lição 29 – Jogos de dois jogadores - tênis para dois



*Este jogo de tênis de dois jogadores, muito mais que a versão com um jogador numa lição anterior, se aproxima ainda mais do jogo clássico originário chamado **Pong**.*

### Condições de realização do jogo - Resumo do plano de codificação

A meta do jogo é que um dos dois jogadores seja declarado vencedor ao fazer com que a bola atinja a linha de cor que está do lado do adversário por cinco vezes. As Raquete actuam ao mesmo tempo como escudos defensivos e como atiradores ofensivos. Cada raquete é controlada por um pacote separado de duas teclas adjacentes (setas Alto / Baixo e W / S).

**Blocos de comandos chave :** « repetir sempre », « Quando recebo e Enviar todos » e « Pára tudo » (pasta Controlo), « Estás a tocar (um actor) » (Sensores), « Alterar \_\_\_ para » e « Adiciona\_\_\_ » (Variáveis), « Número aleatório entre\_\_\_ e\_\_\_ » (Operadores), « Apontar em direcção de\_\_\_ », « Girar » (Movimento).

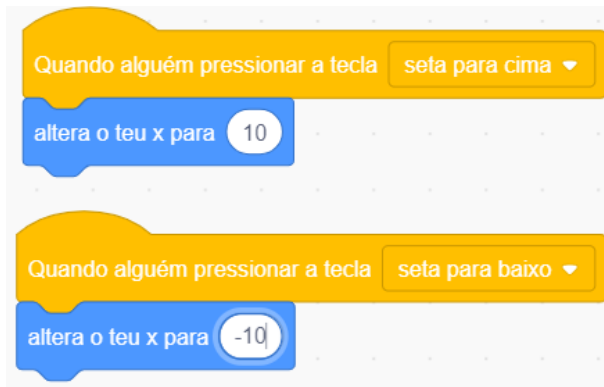
## Guião : actor bola

The image displays eight Scratch code blocks arranged in a 4x2 grid, detailing the logic for a ball actor. Each block starts with a 'Quando alguém clicar em' (When clicked) trigger.

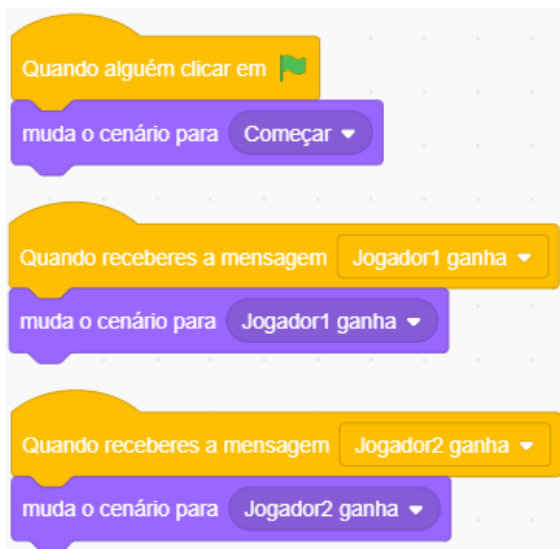
- Block 1 (Top Left):** Sets 'Pontuação Jogador1' and 'Pontuação Jogador2' to 0, then moves to x: 13, y: 157.
- Block 2 (Top Right):** A 'repete para sempre' (forever) loop containing 'se estiveres a bater na borda, ressalta' (if touching the edge, highlight) and 'anda 4 passos' (move 4 steps).
- Block 3 (Second Row Left):** A 'repete para sempre' loop with a 'se estás a tocar em Raquete1' (if touching Raquete1) condition. Inside, it plays a 'Pop' sound, changes direction to 90°, rotates a random angle between 90° and -90°, and moves 12 steps.
- Block 4 (Second Row Right):** A 'repete para sempre' loop with a 'se estás a tocar em Raquete2' (if touching Raquete2) condition. Inside, it plays a 'Pop' sound, changes direction to 90°, rotates a random angle between 90° and -90°, and moves 12 steps.
- Block 5 (Third Row Left):** A 'repete para sempre' loop with a 'se estás a tocar na cor [Red]' (if touching red) condition. It waits 1 second and then adds 1 to 'Pontuação Jogador1'.
- Block 6 (Third Row Right):** A 'repete para sempre' loop with a 'se estás a tocar na cor [Green]' (if touching green) condition. It waits 1 second and then adds 1 to 'Gol'.
- Block 7 (Bottom Row Left):** A 'repete para sempre' loop with a 'se Pontuação Jogador1 = 5' (if Player 1 score is 5) condition. It broadcasts 'Jogador1 ganha' (Player 1 wins), waits 1 second, and then stops everything.
- Block 8 (Bottom Row Right):** A 'repete para sempre' loop with a 'se Pontuação Jogador2 = 5' (if Player 2 score is 5) condition. It broadcasts 'Jogador2 ganha' (Player 2 wins), waits 1 second, and then stops everything.



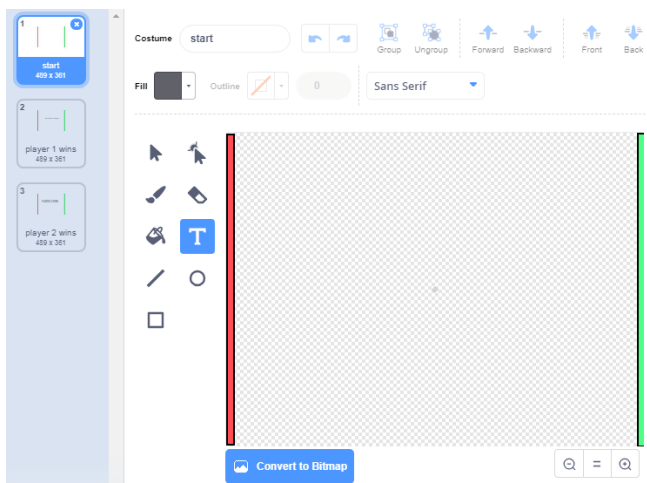
## Guião : actor raquete 1



## Guião : actor palco



Trajes guião palco : a) versão padrão com 2 linhas de golo coloridas nas extremidades opostas ao ecrã, b) o jogador 1 ganha ! e c) O jogador 2 ganha!



## Lição 30 – Quizz de geografia : viajar por um continente

Codificar um **Quizz** com Scratch é uma atividade muito comum nas escolas. As temáticas estendem-se dos conhecimentos gerais até os assuntos específicos. Nessa área, pode dar uma nova dimensão empolgante ao ensino dos idiomas, das ciências, da história e da geografia. Os alunos podem ter a oportunidade de pesquisar as perguntas e as respostas na disciplinas que quiserem. Eles podem também criar um projeto emocionante de **perguntas e respostas** que chamará a atenção e o interesse dos participantes.

O exemplo de projecto abaixo mostrado é aquele dum giro pela Europa. As perguntas estão baseadas aqui no nome dos países. Mas ele poderia ser baseado na designação dos idiomas principais, das capitais, dos chefes de Estado, das destinações turísticas populares, das montanhas, dos rios, dos lagos, etc.



### Condições de realização do jogo - resumo do plano de codificação

O guia chama-se Daire, ele percorre a Europa e pára em cada país para pedir a um jogador que digitem seu nome no ecrã.

A pessoa interrogada não pode deixar o país sem ter digitado a resposta certa. Umavez que a resposta certa for entrada, Daire vai até o próximo país e faz a seguinte pergunta até que todos os países selecionados sejam visitados. Este procedimento continua até que todos os países sejam visitados.

**Blocos de comandos chave :** « repetir para sempre », « Se\_\_então », « Repetir até que\_\_ », « Quando recebo e Enviar a todos » e « Pára tudo » (pasta Controlo), « Diz » (Aparência), « Atingir (uma matiz) », « Pergunta», « Resposta » (Sensores), « Altera \_\_\_ para » e « Adiciona\_\_ » (Variáveis), « \_\_\_ e\_\_ » (Operadores), « Apontar em direcção de\_\_ » (Movimento), « Tocar som\_\_ » (Som).

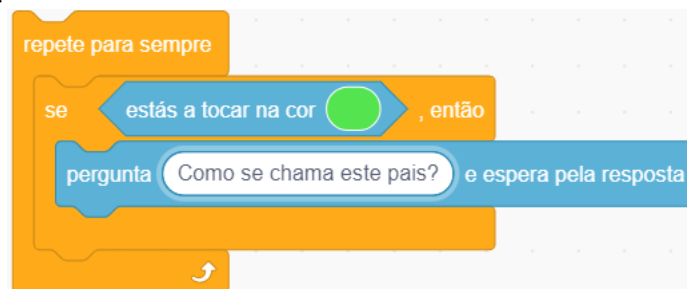
Um Actor (Daire) e um cenário (mapa da Europa) com um único guião para o primeiro.

**Guiões :**

Primeira parte :



Segunda parte:



Terceira parte (incluindo a segunda parte abaixo mostrada)

```

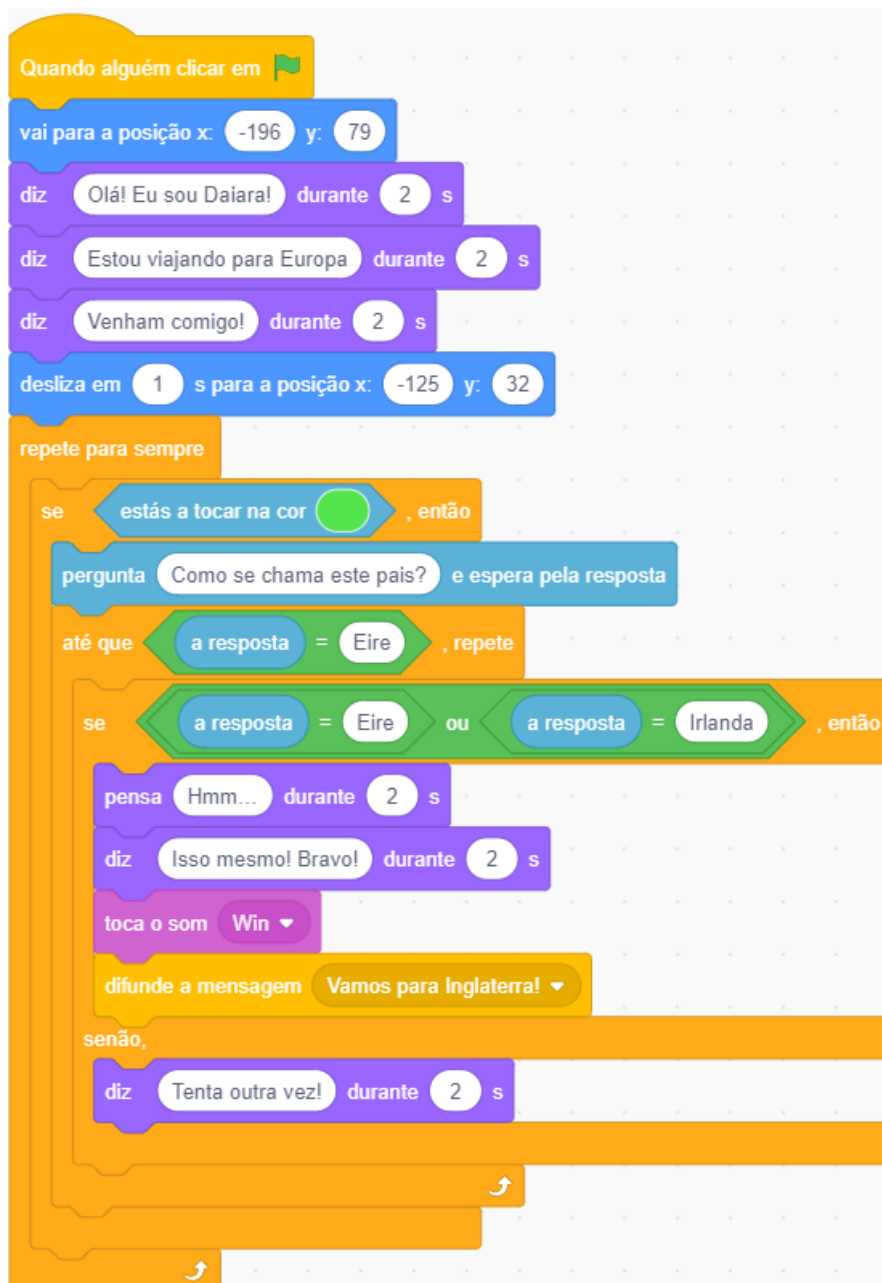
repete para sempre
  se estás a tocar na cor [verde], então
    pergunta Como se chama este país? e espera pela resposta
    se a resposta = Eire ou a resposta = Irlanda, então
      pensa Hmm... durante 2 s
      diz Isso mesmo! Bravo! durante 2 s
      toca o som Win
      difunde a mensagem Vamos para Inglaterra!
    senão,
      diz Tenta outra vez! durante 2 s
  
```

Quarta parte (incluindo a terceira parte acima mostrada)

```

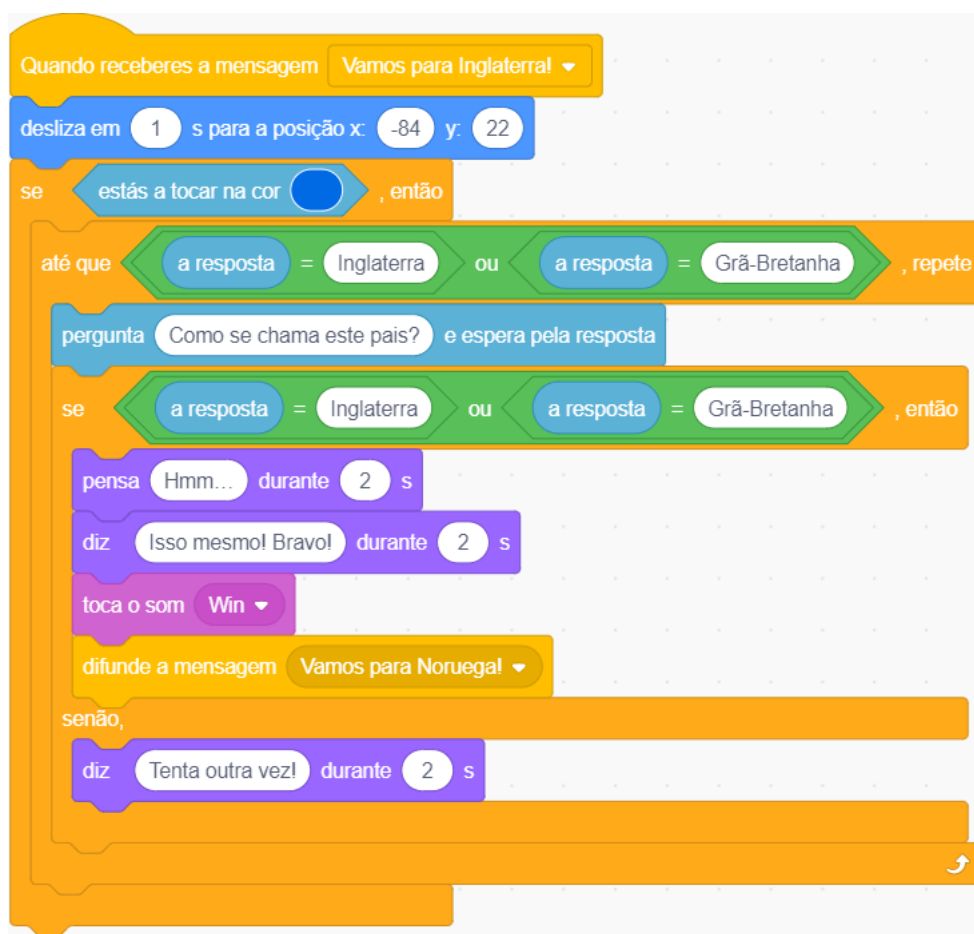
repete para sempre
  se estás a tocar na cor [verde], então
    pergunta Como se chama este país? e espera pela resposta
    até que a resposta = Eire, repete
    se a resposta = Eire ou a resposta = Irlanda, então
      pensa Hmm... durante 2 s
      diz Isso mesmo! Bravo! durante 2 s
      toca o som Win
      difunde a mensagem Vamos para Inglaterra!
    senão,
      diz Tenta outra vez! durante 2 s
  
```

Quarta parte (incluindo as primeira e terceira partes)



Observação : Éire ou Eire é o nome irlandês da Irlanda.

## Guião para o país Inglaterra



Sigam o mesmo procedimento para todos os outros países indicados.

### Exercício

Peçam a todos as crianças da turma ou aos participantes da sessão, que realizem, individualmente ou em grupo, um projecto semelhante baseado no continente africano.

Será preciso pesquisar muito e planejar antecipadamente para conseguir informações pertinentes e necessárias (por exemplo :capitais de países, idiomas falados, características geográficas tais como os rios principais, as montanhas, etc.).

---

<sup>i</sup> Observação : Todas as redações de cursos são preparadas por Brendan Smith, Camden Education Trust, Galway, Irlanda.