

APRENDA A PROGRAMAR COM SCRATCH

**Uma introdução visual à
programação com jogos, arte,
ciência e matemática**

Majed Marji



**no starch
press**

novatec

Copyright © 2014 by Majed Marji. Title of English-language original: *Learn to Program with Scratch*, ISBN 978-1-59327-543-3, published by No Starch Press. Portuguese-language edition copyright © 2014 by Novatec Editora Ltda. All rights reserved.

Copyright © 2014 por Majed Marji. Título original em Inglês: *Learn to Program with Scratch*, ISBN 978-1-59327-543-3, publicado pela No Starch Press. Edição em Português copyright © 2014 pela Novatec Editora Ltda. Todos os direitos reservados.

© Novatec Editora Ltda. 2014.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998. É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates
Tradução: Lúcia Kinoshita
Revisão gramatical: Marta Almeida de Sá
Editoração eletrônica: Carolina Kuwabata

ISBN: 978-85-7522-312-3

Histórico de impressões:

Julho/2014 Primeira edição

Novatec Editora Ltda.
Rua Luís Antônio dos Santos 110
02460-000 – São Paulo, SP – Brasil
Tel.: +55 11 2959-6529
Email: novatec@novatec.com.br
Site: www.novatec.com.br
Twitter: twitter.com/novateceditora
Facebook: facebook.com/novatec
LinkedIn: linkedin.com/in/novatec

1

INTRODUÇÃO AO SCRATCH

Você já sentiu vontade de criar o seu próprio jogo de computador, uma história animada, um tutorial ou uma simulação científica? O *Scratch* é uma linguagem de programação gráfica que facilita a criação dessas aplicações, além de outras tarefas. Neste capítulo introdutório, você irá:

- explorar o ambiente de programação do Scratch;
- conhecer diferentes tipos de blocos de comando;
- criar o seu primeiro jogo no Scratch.

Ao criar uma aplicação no Scratch, você pode salvá-la em seu computador ou pode carregá-la no site do Scratch; nesse último caso, outras pessoas poderão fazer comentários ou transformá-la em novos projetos.

Animado? Então vamos começar!

O que é o Scratch?

Um programa de computador nada mais é do que um conjunto de instruções que dizem o que um computador deve fazer. Essas instruções são escritas por meio de uma *linguagem de programação*, e é aqui que o Scratch entra em cena.

A maioria das linguagens de programação é baseada *em texto*, o que significa que você deve dar comandos ao computador no que parece ser uma forma enigmática de inglês. Por exemplo, para exibir “Hello!” na tela, você pode escrever:

```
print('Hello!')           (na linguagem Python)
std::cout << "Hello!" << std::endl; (na linguagem C++)
System.out.print("Hello!"); (na linguagem Java)
```

Aprender essas linguagens e entender suas regras de sintaxe pode ser um desafio aos iniciantes. O Scratch, por outro lado, é uma linguagem de programação *visual*. Ele foi desenvolvido no Massachusetts Institute of Technology (MIT) Media Lab para tornar o aprendizado de programação mais fácil e mais divertido.

No Scratch, não é necessário digitar nenhum comando complicado. Em vez disso, você irá conectar blocos gráficos para criar programas. Você está confuso? Dê uma olhada no programa simples apresentado na figura 1.1, que eu irei explicar.



Figura 1.1 – Ao executar esse bloco do Scratch, o gato diz “Hello!” em um balão.

O gato que você vê na figura 1.1 é chamado de *sprite* (ator). Os sprites entendem e obedecem a conjuntos de instruções que você lhes atribui. O bloco roxo à esquerda diz ao gato para exibir “Hello!” em um balão. Muitas das aplicações que serão criadas neste livro conterão vários sprites, e você utilizará blocos para fazer com que os sprites se movam, girem, digam algo, toquem música, realizem alguma operação matemática e assim por diante.

Você pode programar em Scratch ao juntar blocos diferenciados por cores, como seria feito com as peças de um quebra-cabeça ou com blocos de LEGO. As pilhas de blocos que forem criadas são chamadas de *scripts* (roteiros). Por exemplo, a figura 1.2 mostra um script que altera a cor de um sprite quatro vezes.

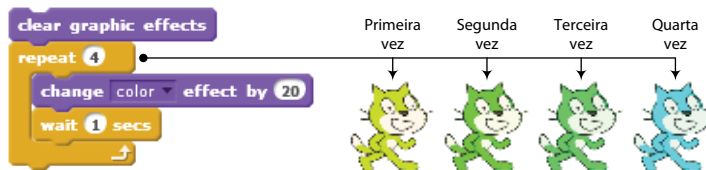


Figura 1.2 – Usando um script para alterar a cor do sprite Cat.

Esse script espera um segundo entre as mudanças de cores, e os quatro gatos que você vê aqui mostram a nova cor do sprite após cada mudança.

Este livro descreve o Scratch 2, disponibilizado em maio de 2013. Essa versão permite criar projetos diretamente em seu navegador web, portanto não é necessário instalar nenhum software em seu computador, e iremos nos basear na interface web do Scratch para o material contido neste livro.

Agora que você já sabe um pouco sobre essa linguagem, é hora de dar início à nossa jornada em programação e aprender a utilizá-la.

O Stage

O Stage (Palco) é o local em que os seus sprites se movem, são desenhados e interagem. O Stage tem 480 passos de largura e 360 passos de altura, conforme mostrado na figura 1.4. O centro do Stage tem coordenada x igual a 0 e coordenada y igual a 0.

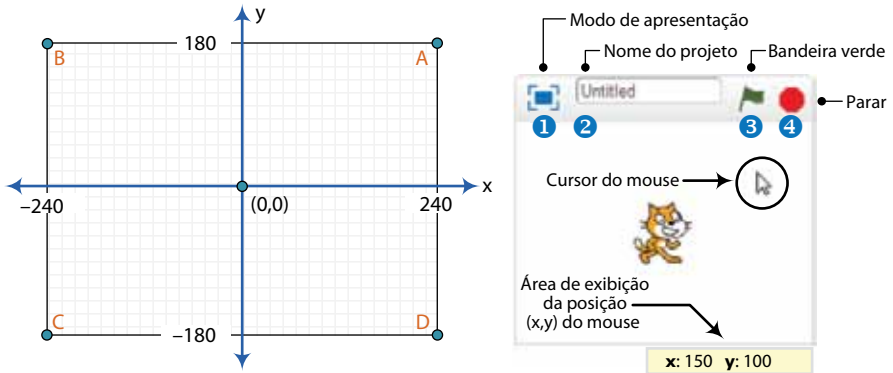



Figura 1.4 – O Stage (Palco) é como um plano de coordenadas com o ponto (0,0) no centro.

As coordenadas (x, y) de qualquer ponto do Stage podem ser encontradas ao mover o cursor do mouse até esse ponto e observar os números na Área de Exibição da Posição (x,y) do Mouse, localizada logo abaixo do Stage.

A pequena barra localizada acima do Stage contém diversos controles. O ícone do modo de Apresentação 1 oculta todos os scripts e as ferramentas de programação e faz com que a área de Stage ocupe quase todo o seu monitor. A caixa de edição 2 mostra o nome do projeto corrente. Os ícones de bandeira verde 3 e de parar 4 permitem iniciar e terminar o seu programa.

FAÇA VOCÊ MESMO 1.2

Mova o mouse pela área de Stage e observe a Área de Exibição da Posição do Mouse. O que acontece quando você move o mouse para fora da área do Stage? Agora mude para o modo de Apresentação e observe como a tela muda. Clique no ícone  na parte superior à esquerda da tela ou teclou esc em seu teclado para sair do modo de Apresentação.

Lista de sprites

A Lista de Sprites (Sprite List) exibe os nomes e as miniaturas (thumbnails) de todos os sprites de seu projeto. Novos projetos começam com o Stage em branco e um único sprite representado por um gato, conforme mostrado na figura 1.5.

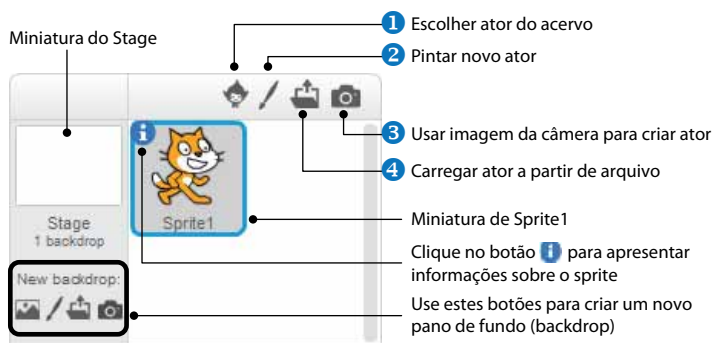


Figura 1.5 – A Lista de Sprites para um novo projeto.

Os botões acima da Lista de Sprites permitem adicionar novos sprites ao seu projeto a partir de um de quatro lugares: da biblioteca de sprites do Scratch ❶, do Paint Editor incluído ❷ (onde você pode desenhar sua própria fantasia), de uma câmera conectada ao seu computador ❸ ou de seu computador ❹.

FAÇA VOCÊ MESMO 1.3

Adicione novos sprites ao seu projeto utilizando alguns dos botões localizados acima da Lista de Sprites. Reorganize os sprites na Lista de Sprites arrastando as miniaturas correspondentes.

Cada sprite de seu projeto tem seus próprios scripts, fantasias e sons. Você pode selecionar qualquer sprite e ver os seus pertences. (1) Clique na miniatura do sprite na Lista de Sprites ou (2) dê um clique duplo no próprio sprite no Stage. A miniatura do sprite selecionado no momento sempre estará em destaque, contornada com uma borda azul. Ao selecionar um sprite, você poderá acessar seus scripts, suas fantasias e os sons ao clicar em uma das três abas localizadas acima da Área de Scripts. Daremos uma olhada nos conteúdos dessas abas posteriormente. Por enquanto, clique com o botão direito do mouse (ou dê um CTRL-clique se você estiver usando um Mac) na miniatura do sprite cat para ver o menu pop-up mostrado na figura 1.6.

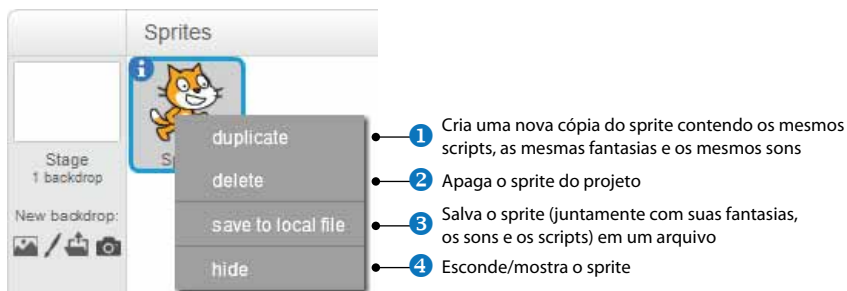


Figura 1.6 – Clicar com o botão direito do mouse em uma miniatura de sprite faz com que esse menu prático seja mostrado.

A opção *duplicate* (duplicar) ❶ copia o *sprite* e dá um nome diferente à cópia. Você pode remover um *sprite* de seu projeto usando a opção *delete* (apagar) ❷, e pode exportar um *sprite* para um arquivo *.sprite2* em seu computador usando a opção *save to local file* (salvar em arquivo local) ❸. [Para importar um *sprite* exportado por outro projeto, basta clicar no botão *Upload sprite from file* (Carregar ator a partir de arquivo) mostrado na figura 1.5]. A opção *hide/show* (esconder/mostrar) ❹ permite alterar a visibilidade de um *sprite* no *Stage*.

Juntamente com as miniaturas de seus *sprites*, a Lista de *Sprites* também mostra uma miniatura do *Stage* à esquerda (veja a figura 1.6). O *Stage* tem seu próprio conjunto de *scripts*, imagens e sons. A imagem de fundo que você vê no *Stage* chama-se pano de fundo (*backdrop*). Ao iniciar um novo projeto, o pano de fundo default do *Stage* é branco, mas você pode adicionar novas imagens de fundo se utilizar qualquer um dos quatro botões abaixo da miniatura do *Stage*. Clique no ícone *Stage* na Lista de *Sprites* para visualizar e editar os *scripts*, os panos de fundo e os sons associados.

A aba *Blocks*

Os blocos no Scratch estão divididos em dez categorias (paletas): *Motion* (Movimento), *Looks* (Aparência), *Sound* (Som), *Pen* (Caneta), *Data* (Variáveis), *Events* (Eventos), *Control* (Controle), *Sensing* (Sensores), *Operators* (Operadores) e *More Blocks* (Mais Blocos). Os blocos são diferenciados por cor para ajudar você a identificar facilmente aqueles que estão relacionados. O Scratch 2 tem mais de cem blocos, embora alguns apareçam somente em determinadas condições. Por exemplo, os blocos da paleta *Data* (Variáveis, discutida nos capítulos 5 e 9) aparecem somente depois que uma variável ou uma lista for criada. Vamos dar uma olhada nos diversos componentes da aba *Blocks* na figura 1.7.

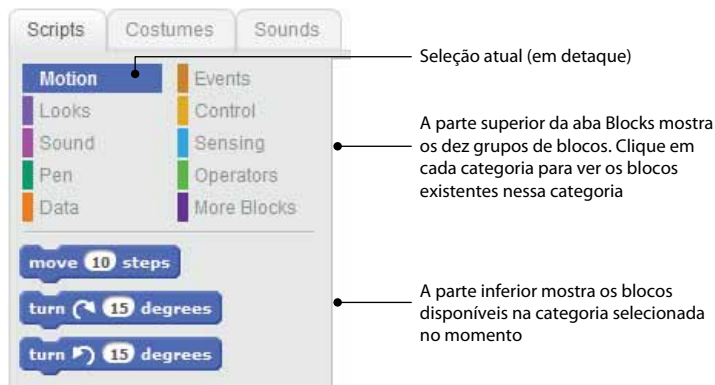


Figura 1.7 – Uma visão ampliada da aba *Blocks* (Blocos).

Experimente clicar em um bloco para ver o que ele faz. Se você clicar em **move 10 steps** (mova 10 passos) da paleta *Motion* (Movimento), por exemplo, o *sprite* deslocará dez passos no *Stage*. Clique nesse bloco novamente e o *sprite* deslocará mais dez passos. Clique no bloco **say Hello! for 2 secs** (diga Hello! por 2 segundos) na paleta *Looks* (Aparência) para fazer o *sprite* mostrar “Hello!” em um balão durante dois segundos. Você também pode acessar a

tela de ajuda de um bloco ao selecionar *Block help* (Ajuda do bloco, que corresponde ao ícone de ponto de interrogação) na barra de ferramentas e clicar no bloco a respeito do qual você tem dúvidas.

Alguns blocos exigem uma ou mais entradas (também chamadas de *argumentos*) que dizem ao bloco o que ele deve fazer. O número 10 no bloco **move 10 steps** (mova 10 passos) é um exemplo de um argumento. Observe a figura 1.8 para ver as diferentes maneiras pelas quais os blocos permitem que suas entradas sejam alteradas.

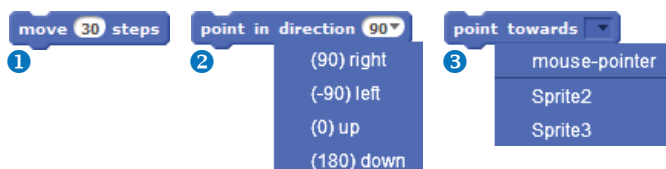


Figura 1.8 – Alterando os dados de entrada de diferentes tipos de blocos.

Você pode alterar o número de passos em **move 10 steps** (mova 10 passos) ao clicar na área branca em que vemos o número 10 e inserir um novo número ①, por exemplo, 30, como pode ser visto na figura 1.8. Alguns blocos, como **point in direction 90** (aponte para a direção 90 graus), também possuem menus suspensos para suas entradas ②. Você pode clicar na seta para baixo para ver uma lista das opções disponíveis e selecionar uma delas. Esse comando em particular apresenta uma área em branco editável, portanto você também pode simplesmente digitar um valor na caixa branca. Outros blocos, como **point towards** (aponte para) ③, forçarão você a escolher um valor a partir do menu suspenso.

FAÇA VOCÊ MESMO 1.4

Vá para a paleta *Looks* (Aparência), altere o valor dos dados de entrada dos blocos e clique neles para ver o que eles fazem. Por exemplo, faça experiências com o bloco **set color effect to** (mude o efeito cor para). Experimente usar números como 10, 20, 30 e assim por diante até que o gato retorne à sua cor original. Experimente usar as opções do menu suspenso com números diferentes. Você pode clicar no bloco **clear graphic effects** [apague os efeitos gráficos, também na paleta *Looks* (Aparência)] para eliminar as suas alterações.

A Área de Scripts

Para fazer com que um sprite faça algo interessante, é necessário programá-lo arrastando blocos da aba *Blocks* (Blocos) para a *Scripts Area* (Área de Scripts), unindo-os. Ao arrastar um bloco para a Área de Scripts, uma área destacada em branco indica em que local esse bloco pode ser solto para formar uma conexão válida com outro bloco (Figura 1.9). Os blocos do Scratch somente se encaixam de determinadas maneiras, eliminando os erros de digitação que tendem a ocorrer quando as pessoas usam linguagens de programação baseadas em texto.

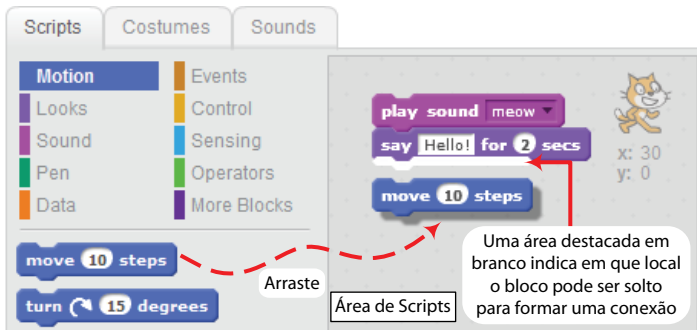
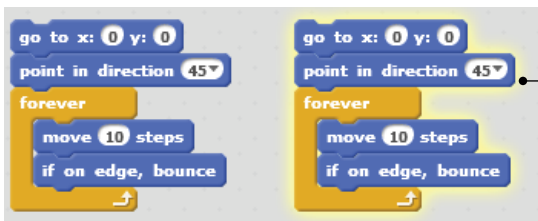


Figura 1.9 – Arraste os blocos para a Área de Scripts e una-os para criar scripts.

Não é preciso completar os scripts para executá-los, o que significa que você pode testá-los à medida que os criar. Clicar em qualquer ponto de um script, esteja ele totalmente ou parcialmente criado, faz o script todo ser executado de cima para baixo.

FAÇA VOCÊ MESMO 1.5

Inicie um novo projeto no Scratch e crie o script abaixo para o sprite Cat. [O bloco **forever** (sempre) está na paleta Control (Controle), e os demais blocos estão na paleta Motion (Movimento).]



Esta borda amarela indica que o script está sendo executado

Você conhecerá a maior parte desses blocos no capítulo 2. Por enquanto, clique em seu novo script para executá-lo. (O Scratch deve deixar o script em execução em destaque com uma borda amarela brilhante, como mostrado na imagem à direita.) Você pode até mesmo alterar os dados de entrada de um bloco e adicionar blocos novos a um script enquanto ele estiver executando! Por exemplo, altere o número do bloco **move** (mova) e observe como o movimento do gato se altera. Clique no script mais uma vez para interrompê-lo.

Você pode facilmente desmontar uma pilha de blocos e testar cada uma das partes individualmente. Essa será uma estratégia de valor inestimável quando você estiver tentando entender scripts longos. Para mover toda uma pilha de blocos, clique no bloco superior da pilha. Para desassociar um bloco que está no meio de uma pilha e todos os blocos abaixo dele, clique nesse bloco e arraste-o. Vá em frente e experimente fazer isso.

Esse recurso também permite criar o seu projeto por partes. Você pode conectar conjuntos pequenos de blocos, testá-los para garantir que funcionam conforme esperado e então combiná-los em scripts mais longos.

Você pode até mesmo copiar uma pilha de blocos de um sprite para outro. Basta arrastar a pilha da Área de Scripts do sprite de origem para a miniatura do sprite de destino na Lista de Sprites.

FAÇA VOCÊ MESMO 1.6

Adicione outro sprite ao seu projeto. Arraste o script do sprite Cat e solte-o sobre a miniatura do novo sprite. O ponteiro do mouse deve estar sobre a miniatura do novo sprite para que esse processo seja bem-sucedido. Verifique a aba Scripts (Roteiros) do novo sprite para garantir que ele contenha uma cópia idêntica do script.

A aba Costumes

A aparência de um sprite pode ser alterada se mudarmos a sua fantasia (costume), que é somente uma imagem. A aba Costumes (Fantasias) contém tudo o que é necessário para organizar as fantasias de seu sprite; você pode imaginá-la como se fosse um guarda-roupa. O guarda-roupa pode conter diversas fantasias, porém um sprite pode vestir somente uma em um determinado instante.

Vamos agora tentar alterar a fantasia do sprite cat. Clique na miniatura do sprite cat e selecione a aba Costumes (Fantasias). Como mostrado na figura 1.10, Cat tem duas fantasias: costume1 e costume2. A fantasia em destaque (costume1, nesse caso) representa a fantasia atual do sprite.

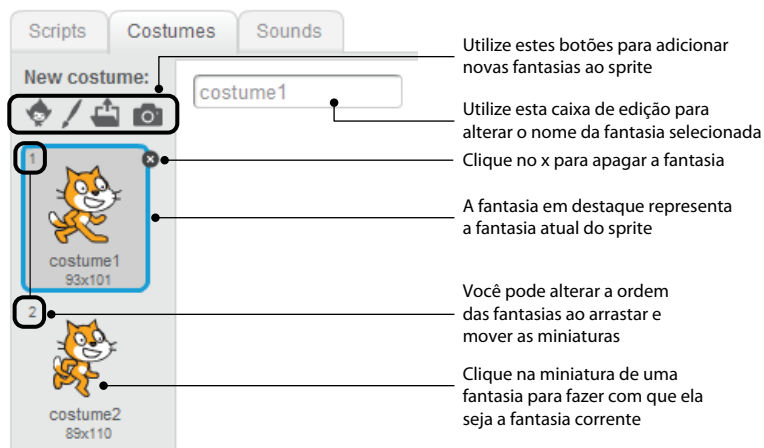


Figura 1.10 – Você pode organizar todas as fantasias de um sprite a partir da aba Costumes (Fantasias).

Se você clicar com o botão da direita do mouse na miniatura de uma fantasia, um menu pop-up com três opções será apresentado: (1) duplicate (duplicar), (2) delete (apagar) e (3) save to local file (salvar em arquivo local). A primeira opção adiciona uma nova fantasia com uma imagem idêntica à da fantasia que você estiver duplicando. A opção delete apaga a fantasia selecionada. A última opção permite salvar a fantasia em um arquivo. Você pode

importar essa fantasia e utilizá-la em um projeto diferente ao usar o botão Upload costume from file (Carregar fantasia a partir de arquivo, que é o terceiro botão na figura 1.10). Vá em frente e experimente usar essas opções.

FAÇA VOCÊ MESMO 1.7

Clique no primeiro botão acima da imagem do gato na figura 1.10 para selecionar uma nova fantasia a partir da biblioteca do Scratch. Em seguida, selecione qualquer imagem que você gostar na janela que for apresentada. Aplique algumas das dicas que estão na figura 1.10 para se familiarizar mais com as opções associadas às fantasias.

A aba Sounds

Os sprites também podem reproduzir sons, o que deixa os seus programas mais animados. Por exemplo, você pode disponibilizar diferentes sons a um sprite para que sejam usados quando ele estiver feliz ou triste. Se o seu jogo contiver um sprite que se pareça com um míssil, você poderá fazê-lo gerar sons diferentes de acordo com o fato de o míssil atingir ou errar um alvo.

Os botões da aba Sounds (Sons) ajudarão a organizar os diferentes sons que seus sprites podem reproduzir. Como mostrado na figura 1.11, o Scratch oferece até mesmo uma ferramenta que pode ser usada para editar arquivos de áudio. Não discutirei os detalhes a respeito dessa ferramenta neste livro, porém sugiro que você faça experimentos por conta própria.

Na maioria das vezes, você precisará somente dos três botões que estão na parte superior da aba Sounds. Eles permitem selecionar um som a partir da biblioteca de sons do Scratch ❶, gravar um áudio novo ❷ (se você tiver um microfone) ou importar um arquivo de áudio de seu computador ❸. O Scratch pode ler somente arquivos de áudio MP3 e WAV.



Figura 1.11 – A aba Sounds (Sons) permite organizar os sons associados a um sprite.

FAÇA VOCÊ MESMO 1.8

Selecione a aba Sounds (Sons) e clique no botão **Choose sound from library** (Escolher som do acervo). Ouça os vários sons que estão disponíveis no Scratch para ter algumas ideias para seus projetos futuros.

A aba Backdrops

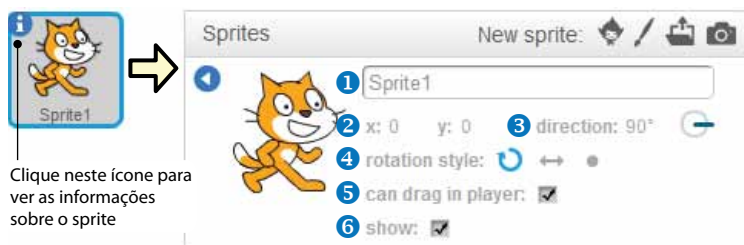
Ao selecionar a miniatura do Stage na Lista de Sprites, o nome da aba do meio mudará de *Costumes* (Fantasias) para *Backdrops* (Panos de fundo). Utilize essa aba para organizar as imagens de fundo do Stage, que podem ser alteradas juntamente com seus scripts. Por exemplo, se você estiver criando um jogo, um pano de fundo contendo instruções para começar poderá ser apresentado e, em seguida, poderá ser alterado para algo diferente quando o usuário iniciar o jogo. A aba Backdrops (Panos de fundo) é idêntica à aba Costumes (Fantasias).

FAÇA VOCÊ MESMO 1.9

Clique no botão **Choose backdrop from library** (Escolher fundo do acervo) abaixo da miniatura do Stage na Lista de Sprites. Selecione o pano de fundo xy-grid na janela apresentada e clique em OK. O Scratch adicionará o xy-grid ao seu projeto e fará com que ele seja o pano de fundo default. (O xy-grid mostra um plano cartesiano de duas dimensões, que será útil quando você estiver trabalhando com os blocos de comando de Movimento.) Repita esses passos e selecione qualquer outro pano de fundo que você desejar.

Informações sobre o sprite

Você pode visualizar a área de informações sobre o sprite ao clicar no pequeno ícone ⓘ no canto superior esquerdo da miniatura de um sprite, como mostrado na figura 1.12. Essa área mostra o nome do sprite, sua posição atual (x,y) e a direção, seu estilo de rotação e o estado quanto à visibilidade, além de mostrar se ele pode ser arrastado em modo de Apresentação. Vamos discutir brevemente cada uma dessas opções.



Clique neste ícone para ver as informações sobre o sprite

Figura 1.12 – A área de informações do sprite.

A caixa de edição ❶ na parte superior dessa área permite alterar o nome do sprite. Você utilizará essa caixa diversas vezes neste livro.

Os valores de x e de y ❷ mostram a posição atual do sprite no Stage. Arraste o sprite pelo Stage e observe o que acontece com esses números.

A direção do sprite ❸ indica para qual direção o sprite irá se mover em resposta a um bloco de movimento. Arraste a linha azul que parte do centro do ícone em forma de círculo para efetuar uma rotação no sprite.

Os três botões de estilo de rotação ❹ [que se chamam Rotate (Efetuar rotação), Left-right flip (Virar à esquerda e à direita) e No rotate (Sem rotação)] controlam o modo como a fantasia aparecerá à medida que o sprite mudar a sua direção. Para compreender o efeito desses botões, crie o sprite mostrado na figura 1.13 e clique em cada um desses botões enquanto o script estiver executando. O bloco **wait** (espere) pode ser encontrado na paleta *Control* (Controle).

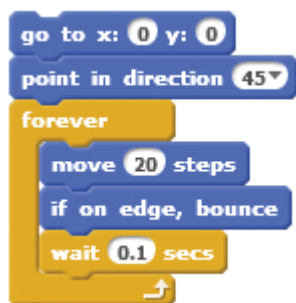


Figura 1.13 – Script para demonstrar os estilos de rotação.

A caixa de seleção Can drag in player (Pode ser arrastado) ❺ indica se o sprite pode ou não ser arrastado (usando o mouse) em modo de Apresentação. Mude para o modo de Apresentação com essa caixa selecionada/não selecionada e tente arrastar o sprite pelo Stage para compreender o efeito dessa caixa de seleção.

A caixa de seleção Show (Mostrar) ❻ permite mostrar/ocultar o sprite durante o design do programa. Experimente usar essa opção para ver o que acontece. Você verá diversos exemplos de sprites ocultos que realizam tarefas úteis por trás das cortinas em vários exemplos ao longo deste livro.

A barra de ferramentas

Vamos dar uma olhada rápida na barra de ferramentas do Scratch na figura 1.14, começando por alguns dos botões. (A barra de ferramentas terá uma aparência um pouco diferente se você estiver logado, conforme será discutido no Apêndice A.) Utilize os botões Duplicate (Duplicar) e Delete (Apagar) para copiar e remover sprites, fantasias, sons, blocos ou scripts. O botão Grow (Aumentar) aumenta o sprite, enquanto o botão Shrink (Reduzir) faz com que ele diminua. Basta clicar no botão que você quiser usar e, em seguida, clicar em um sprite (ou em um script) para aplicar essa ação. Para retornar ao cursor em forma de seta, clique em qualquer área em branco da tela. Você pode utilizar o menu Language (Idiomas) para alterar o idioma da interface do usuário.

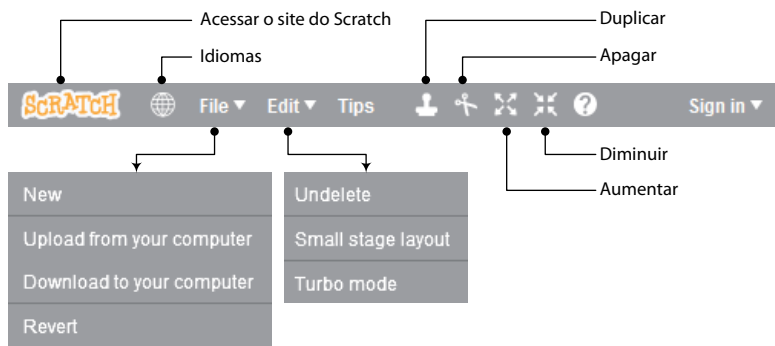


Figura 1.14 – A barra de ferramentas do Scratch.

A partir do menu File (Arquivo), você pode criar novos projetos, carregar (abrir) um projeto existente em seu computador, descarregar (salvar) o projeto corrente para o seu computador ou desfazer (undo) todas as alterações feitas no projeto corrente. Os projetos do Scratch 2 têm uma extensão de arquivo `.sb2` para distingui-los dos projetos criados com a versão anterior do Scratch (`.sb`).

No menu Edit (Editar), a opção Undelete (Recuperar) trará de volta o último bloco, script, sprite, fantasia ou som que você apagou. A opção Small stage layout (Disposição com palco pequeno) diminui o Stage e proporciona mais espaço para a Área de Scripts. Selecionar Turbo mode (Modo turbo) aumenta a velocidade de alguns blocos. Por exemplo, executar um bloco **move** (mova) 1.000 vezes pode levar cerca de 70 segundos em modo normal e aproximadamente 0,2 segundo em modo Turbo.

Agora que vimos o essencial sobre a barra de ferramentas do Scratch, discutiremos brevemente o Paint Editor incluído no Scratch.

Paint Editor

Você pode usar o Paint Editor (Figura 1.15) para criar ou editar fantasias (costumes) e panos de fundo (backdrops). (É claro que você também pode usar o seu programa favorito de edição de imagens.) Se quiser aprender mais sobre o Paint Editor do Scratch, dê uma olhada em *ScratchPaintEditor.pdf* (localizado junto aos recursos online, que podem ser baixados a partir de <http://nostarch.com/learnscratch/>).

Por enquanto, há dois recursos importantes que você deve conhecer: definir o centro de uma imagem e definir a cor transparente. Vou explicar esses recursos nas seções a seguir.

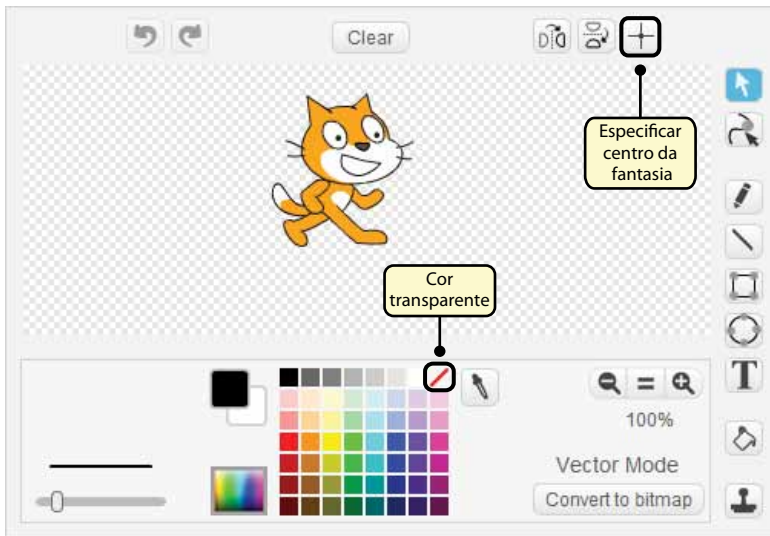


Figura 1.15 – O Paint Editor do Scratch.

Definindo o centro de uma imagem

Ao dar um comando para que um sprite vire (para a esquerda ou para a direita), ele irá girar em relação a um ponto de referência – o centro de sua fantasia. O botão Set costume center (Especificar centro da fantasia, no canto superior direito do Paint Editor) permite selecionar esse centro. Ao clicar nesse botão, você verá uma cruz na área de desenho, conforme mostrado na figura 1.16. O ponto central é determinado pela intersecção desses dois eixos, portanto, para alterar o centro de uma fantasia, basta arrastá-los para uma nova posição. Para ocultar os eixos, clique no mesmo botão novamente.

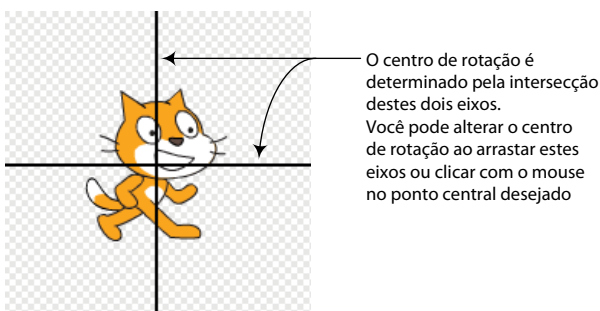
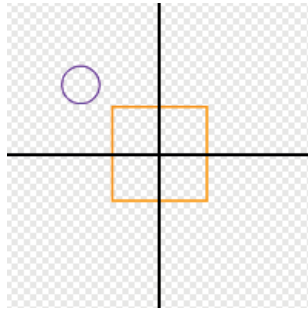


Figura 1.16 – Alterando o centro de uma fantasia após clicar no botão Set costume center (Especificar centro da fantasia).

FAÇA VOCÊ MESMO 1.10

RotationCenter.sb2

Abra o *RotationCenter.sb2* e execute-o. Essa aplicação contém um único sprite com a fantasia e o script mostrados abaixo. O centro da fantasia está definido no meio do quadrado. Execute o script e observe o padrão. Em seguida, altere a fantasia de modo a definir o seu centro no meio do círculo e execute o script novamente para ver como a figura se altera.



```
go to x: 0 y: 0
point in direction 90
clear graphic effects
clear
forever
  turn 5 degrees
  change color effect by 1
  stamp
```

Definindo a cor transparente

Quando duas imagens se sobrepõem, a imagem de cima cobrirá algumas partes da imagem de baixo. De modo semelhante, os sprites cobrem partes do Stage. Se você quiser ver a aparência do Stage por trás de uma imagem, será necessário usar o Paint Editor para deixar pelo menos parte da imagem *transparente*, como o pinguim que está à direita na figura 1.17.

Na paleta *Color* (Cores), basta clicar no quadrado com uma linha vermelha na diagonal e pintar com essa cor “transparente” para deixar algo invisível. Você pode pensar nesse ícone como um sinal de “Ausência de cor”, semelhante a um sinal de “Proibido fumar” com uma barra vermelha sobre um cigarro.

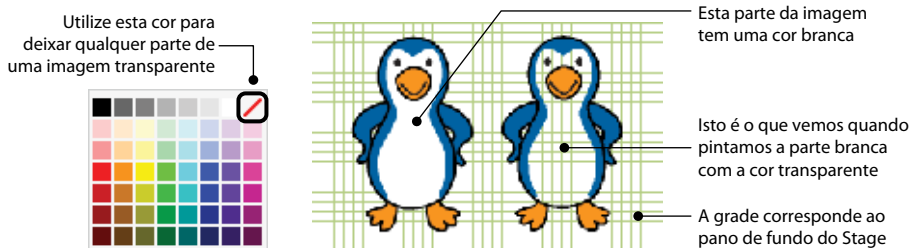


Figura 1.17 – Você pode deixar qualquer parte de uma imagem transparente ao preenchê-la com a cor “transparente”.

Agora que você conhece a interface do Scratch, faremos um bom uso desse conhecimento e criaremos algo divertido. Arregace as mangas e prepare-se: vamos criar um jogo!

Seu primeiro jogo com o Scratch

Pong.sb2

Pong_NoCode.sb2

Nesta seção, você criará um jogo para um só participante, no qual os jogadores deslocarão uma raquete para evitar que uma bola de tênis atinja o piso, de acordo com o clássico jogo Pong do tipo arcade. A interface de usuário para o nosso jogo está sendo mostrada na figura 1.18.

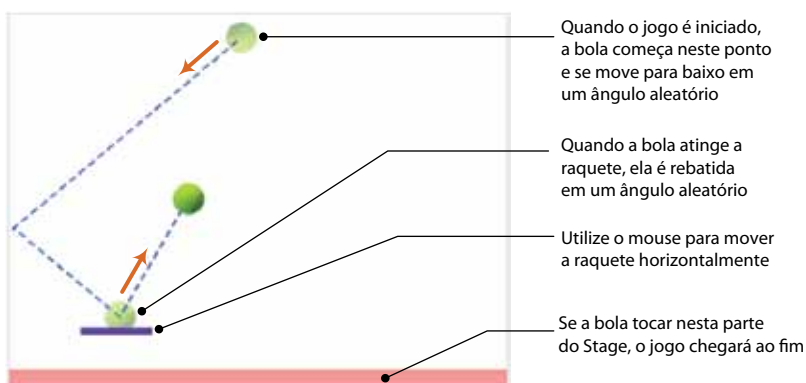


Figura 1.18 – A tela de nosso jogo.

Como mostrado na figura, a bola começa na parte superior do Stage e se move para baixo em um ângulo aleatório, sendo rebatida ao tocar nas bordas do Stage. O jogador move a raquete horizontalmente (usando o mouse) para enviar a bola de volta para cima. Se a bola tocar a parte inferior do Stage, o jogo chegará ao fim.

Criaremos esse jogo passo a passo, porém, inicialmente, devemos criar um novo projeto. Selecione **File ▶ New** (Arquivo ▶ Novo) para iniciar um novo projeto no Scratch. Em seguida, apague o sprite cat ao clicar no botão direito do mouse e selecionar **delete** (apagar) no menu pop-up.

Passo 1: Preparar o pano de fundo

Para detectar se a bola não foi rebatida pela raquete, marcaremos a parte inferior do Stage com uma determinada cor e utilizaremos o bloco **touching color ?** [tocando na cor ?, da paleta *Sensing* (Sensores)] para nos informarmos quando a bola tocar nessa cor. Nosso pano de fundo (backdrop) atual é branco, portanto podemos simplesmente desenhar um retângulo estreito e colorido na parte inferior, conforme mostrado na figura 1.19.

Clique na miniatura do Stage para selecioná-lo e, em seguida, acesse a aba Backdrops (Panos de fundo). Siga os passos da figura 1.19 para desenhar um retângulo estreito na parte inferior do pano de fundo do Stage.



Figura 1.19 – Passos para desenhar um retângulo na parte inferior da imagem do pano de fundo.

Passo 2: Adicionar a raquete e a bola

Clique no botão **Paint new sprite** (Pintar novo ator) acima da Lista de Sprites para adicionar o sprite `Paddle` ao seu projeto. Como a raquete é somente um retângulo pequeno e estreito, repita o que foi feito no Passo 1 para desenhar uma raquete como a que está sendo mostrada na figura 1.18. Dê a cor que quiser à raquete e defina o centro aproximadamente no meio do retângulo.

A seguir, dê um nome ao sprite que explique o que ele faz; eu o chamei de `Paddle`. Além disso, clique na imagem da raquete no Stage e mova-o para que a sua coordenada `y` seja aproximadamente igual a `-120`.

O nosso jogo agora tem uma raquete, porém ainda precisamos de uma bola para bater, portanto clique em **Choose sprite from library** (Escolher ator do acervo) acima da Lista de Sprites para importar um. No diálogo que for apresentado, clique na categoria **Things** (Coisas) e selecione a imagem **Tennis Ball** para adicionar esse sprite ao seu projeto. Renomeie o sprite como `Ball`.

Antes de começar a trabalhar nos scripts para o jogo, selecione **File ▶ Download to your computer** (Arquivo ▶ Baixar para o seu computador) para salvar o que você fez até agora em seu computador. No diálogo que for apresentado, selecione a pasta em que você quer salvar o seu trabalho, chame o arquivo de `Pong.sb2` e clique em **Save** (Salvar). Se estiver logado no momento, você também poderá salvar o seu trabalho na *nuvem* (ou seja, no servidor do Scratch). Independentemente de você ter decidido salvar seus arquivos localmente (em seu computador) ou na nuvem, não se esqueça de salvar o seu trabalho com frequência.

Com os sprites `Paddle` e `Ball`, o Stage deverá ter uma aparência semelhante à da figura 1.18. Se você tiver dificuldades nesse momento, abra o arquivo `Pong_NoCode.sb2`, que contém tudo o que acabamos de criar. Você adicionará os scripts para executar o jogo a seguir, mas não se preocupe demais com os detalhes dos blocos. Iremos explorar todos eles posteriormente no livro, portanto, por enquanto, vamos focar em aprender a criar um projeto completo.

Passo 3: Iniciar o jogo e fazer seus sprites se moverem

Como designer desse jogo, você decidirá como os jogadores podem iniciar uma nova partida. Por exemplo, o jogo pode começar quando você pressionar uma tecla, clicar em um sprite no Stage ou até mesmo quando você bater palmas ou acenar (se você tiver uma webcam). O ícone de bandeira verde (localizado acima do Stage) é outra opção popular que será utilizada nesse caso.

A ideia é simples. Qualquer script que iniciar com o bloco de trigger **when green flag clicked** (quando clicar em bandeira verde) iniciará a execução quando esse botão for pressionado. A bandeira assumirá a cor verde brilhante e permanecerá assim até o script terminar. Para ver isso em ação, crie o script que está sendo mostrado na figura 1.20 para o sprite Paddle.

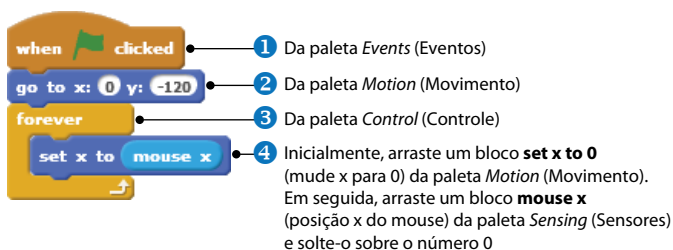


Figura 1.20 – O script para o sprite Paddle.

Quando a bandeira verde for clicada ①, o bloco **go to x: y:** (vá para x: y:) ② definirá a posição vertical da raquete para -120, somente no caso de você tê-la movido anteriormente com o mouse. A raquete deve flutuar imediatamente acima do retângulo rosa que está na parte inferior do Stage, portanto, se o seu retângulo for mais largo, altere o número referente à sua posição para algo que funcione em seu design.

O script então utiliza um bloco **forever** (sempre) ③ para verificar constantemente a posição do mouse. Iremos mover a raquete para trás e para a frente ao fazer a correspondência da posição *x* da raquete com essa posição do mouse ④. Execute o script (clicando no ícone da bandeira verde) e experimente mover o seu mouse horizontalmente; a raquete deverá seguir o movimento. Clique no ícone de parar ao lado da bandeira verde para interromper o script.

O script para o sprite *Ball* é um pouco mais longo que o anterior e, sendo assim, vou dividi-lo em porções mais simples. A bola deve começar a se mover quando clicarmos na bandeira verde, portanto, inicialmente, adicione o script da figura 1.21 ao sprite *Ball*.

Inicialmente, iremos mover a bola para a parte superior do palco ① e faremos com que ela aponte para baixo em um ângulo aleatório usando o bloco **pick random** (escolha número entre) ② [da paleta *Operators* (Operadores)]. O script então utiliza um bloco **forever** (sempre) ③ para mover a bola ④ pelo Stage e fazê-la ser rebatida ⑤ se tocar nas bordas. Clique na bandeira verde para testar o que foi implementado até agora. A bola deve mover-se em um padrão de zig-zaque e a raquete deve continuar seguindo o mouse.

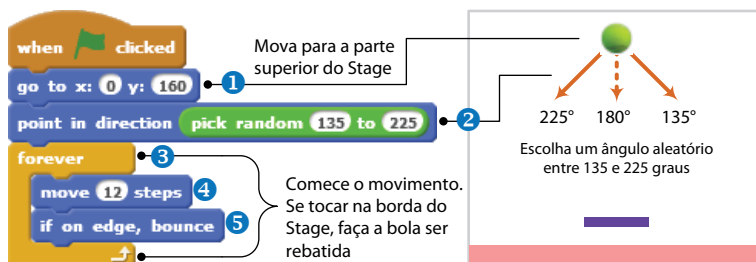


Figura 1.21 – A primeira parte do script para o sprite Ball.

FAÇA VOCÊ MESMO 1.11

Substitua o 12 dentro do bloco **move** (mova) por valores diferentes, execute o script e observe o que acontece. Isso deverá dar a você uma ideia de como deixar o jogo mais fácil ou mais difícil. Clique no ícone de parar quando terminar.

Agora é hora de acrescentar a parte divertida – os blocos que fazem a bola ser rebatida pela raquete. Podemos modificar o bloco **forever** (sempre) que acabamos de criar para que a bola vá para cima quando atingir a raquete, conforme mostrado na figura 1.22.

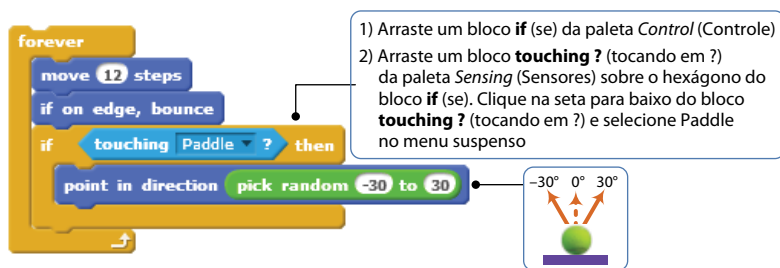


Figura 1.22 – Adicionando código para a bola subir.

Quando a bola e a raquete se tocarem, ordenaremos à bola para que aponte em uma direção aleatória entre -30 e 30 . Quando o bloco **forever** (sempre) executar a próxima rodada, o bloco **move** (mova) será executado, o que fará a bola ir para cima. Clique na bandeira verde novamente para testar essa parte do jogo. Clique no ícone de parar quando tiver certeza de que a bola está sendo rebatida pela raquete conforme esperado.

A única parte que está faltando agora é um código que faça o jogo parar quando a bola tocar a parte inferior do Stage. Adicione o script que está sendo mostrado na figura 1.23 ao sprite Ball, imediatamente antes ou depois do bloco **if** (se) da figura 1.22. Você encontrará o bloco **touching color ?** (tocando na cor ?) na paleta *Sensing* (Sensores) e o bloco **stop** (pare) na paleta *Control* (Controle).

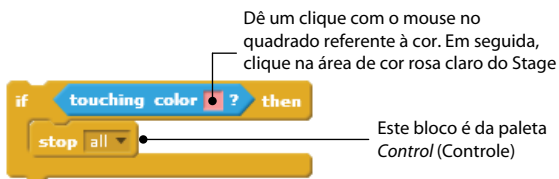


Figura 1.23 – Os blocos para finalizar o jogo.

Ao clicar com o mouse no quadrado colorido dentro do bloco **touching color ?** (tocando na cor ?), o cursor será alterado para a imagem de uma mão. Ao mover esse cursor e clicar sobre o retângulo rosa-claro na parte inferior do Stage, o quadrado colorido dentro do bloco deverá ficar com a mesma cor do retângulo. O bloco **stop all** (pare todos) faz exatamente o que o seu nome diz: todos os scripts em execução em todos os sprites são interrompidos, e os sprites `Paddle` e `Ball` não constituem nenhuma exceção.

Esse jogo básico de pong agora está totalmente funcional. Clique na bandeira verde e jogue algumas vezes para testá-lo. Após ter percebido que é possível criar um jogo inteiro com uma quantidade tão pequena de código, espero que você concorde comigo que o Scratch é incrível!

Passo 4: Dar uma animada com sons

É claro que os jogos são muito mais divertidos quando têm som, portanto vamos adicionar um toque final de modo que um som seja reproduzido sempre que acertarmos a bola.

Dê um clique duplo na bola que está no Stage para selecioná-la e, em seguida, selecione a aba Sounds (Sons). Clique no botão **Choose sound from library** (Escolher som do acervo) para adicionar um som ao sprite `Ball`. No diálogo que for apresentado, selecione a categoria **Effects** (Efeitos), selecione o som **pop** e clique em **OK** para adicioná-lo à aba Sounds (Sons). Depois disso, retorne à aba Scripts (Roteiros) e insira um bloco **play sound** [toque o som, da paleta *Sound* (Som)], como mostrado na figura 1.24.

Teste o jogo uma vez mais e, dessa vez, você deverá ouvir um breve som “pop” sempre que a bola tocar na raquete.

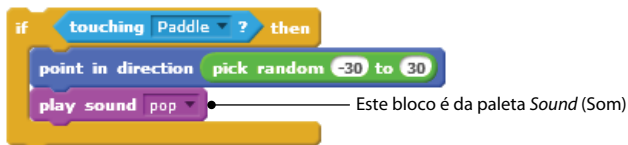


Figura 1.24 – Reproduzindo um som quando a bola tocar na raquete.

Parabéns! Seu jogo agora está completo (a menos, é claro, que você queira acrescentar mais recursos a ele), e você acabou de implementar o seu primeiro programa no Scratch. Se quiser realizar mais alguns experimentos, tente duplicar o sprite `Ball` para ter duas (ou mais) bolas em seu jogo e veja como isso muda a maneira de jogar!

Na próxima seção, vou apresentar os diferentes tipos de blocos disponíveis no Scratch. À medida que prosseguir neste livro, você realmente irá explorar a maneira como esses blocos funcionam, mas, por enquanto, iremos descrevê-los apenas brevemente.

Blocos do Scratch: uma visão geral

Nesta seção, você conhecerá os diferentes tipos de blocos disponíveis no Scratch, seus nomes e os usos pretendidos. O objetivo é definir alguns dos termos que serão vistos nos próximos capítulos. Você pode retornar a esta seção à medida que progredir, se for necessário refrescar a sua memória.

Como está sendo mostrado na figura 1.25, o Scratch possui quatro tipos de blocos: blocos de comando, blocos de função, blocos de trigger e blocos de controle. *Os blocos de comando e os blocos de controle (também chamados de blocos de pilha) têm saliências na parte inferior e/ou nas reentrâncias na parte superior.* Esses blocos podem ser unidos na forma de pilhas. *Os blocos de trigger, também chamados de hats (chapéus), têm a parte superior arredondada porque são colocados no topo de uma pilha.* Os blocos de trigger conectam eventos a scripts. Eles esperam por um evento – por exemplo, um pressionamento de tecla ou um clique do mouse – e executam os blocos que estão abaixo deles quando esse evento ocorre. Por exemplo, todos os scripts que começarem com o bloco **when green flag clicked** (quando clicar em bandeira verde) serão executados quando o usuário clicar no ícone da bandeira verde.

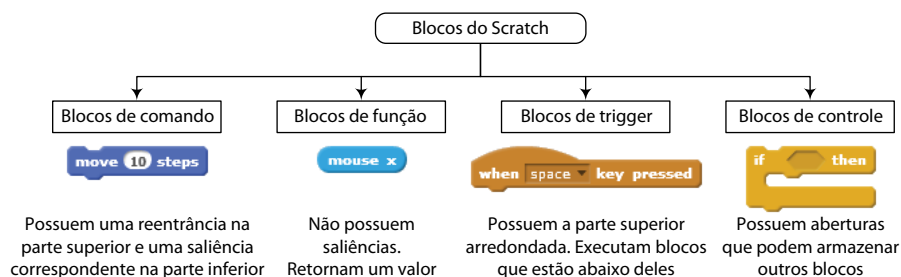


Figura 1.25 – Os quatro tipos de blocos disponíveis no Scratch.

Os blocos de função [também chamados de *reporters (informantes)*] não têm saliências nem reentrâncias. Eles não podem constituir uma camada de um script sozinhos; em vez disso, são usados como entradas para outros blocos. Os formatos desses blocos indicam o tipo de dado que eles retornam. Por exemplo, os blocos com extremidades arredondadas informam números ou strings, enquanto os blocos com as extremidades pontiagudas informam se algo é verdadeiro ou falso. Isso está sendo mostrado na figura 1.26.

Alguns blocos de função apresentam uma caixa de seleção ao lado deles. Se a caixa for selecionada, um *monitor* aparecerá no Stage para mostrar o valor corrente do bloco informante. Selecione um sprite e marque a caixa de seleção do bloco **x position** [coordenada x da posição, da paleta *Motion (Movimento)*]. Em seguida, arraste o sprite pelo Stage e observe esse monitor. Ele deverá ser alterado à medida que o sprite for movido para trás e para a frente.

x position Blocos de função com extremidades arredondadas informam números ou strings

mouse down? Blocos de função com extremidades pontiagudas informam valores booleanos (verdadeiro/falso)

Figura 1.26 – O formato de um bloco de função indica o tipo de dado que ele retorna.

Operadores aritméticos e funções

Agora vamos dar uma olhada rápida nos operadores aritméticos e nas funções suportadas pelo Scratch. Se você perdeu sua calculadora, então seus problemas acabaram! Você pode criar a sua própria calculadora no Scratch usando os blocos da paleta *Operators* (Operadores), que serão explorados nesta seção.

Operadores aritméticos

O Scratch suporta as quatro operações aritméticas básicas de adição (+), subtração (-), multiplicação (*) e divisão (/). Os blocos usados para realizar essas operações, chamados de *operadores*, estão sendo mostrados na figura 1.27. Como esses blocos geram um número, você pode utilizá-los como entradas para qualquer bloco que aceite números, conforme mostrado nessa figura.

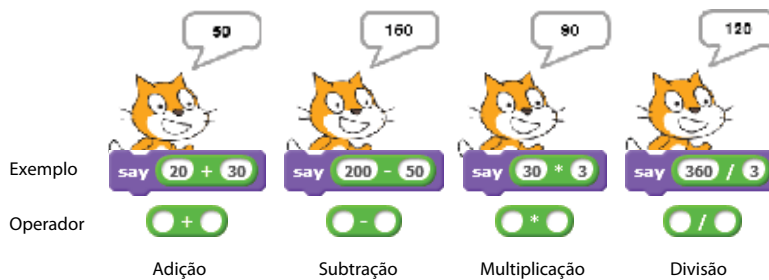


Figura 1.27 – Operadores aritméticos do Scratch.

O Scratch também suporta o operador de módulo (**mod**), que retorna o resto da divisão de dois números. Por exemplo, **10 mod 3** (resto de 10 por 3) retorna 1 porque o resto da divisão de 10 por 3 é 1. Um uso comum do operador de módulo está em testar a divisibilidade de um *inteiro* por outro inteiro (menor). Um módulo igual a 0 indica que o número maior é divisível pelo número menor. Isso dá a você alguma ideia para verificar se um número é par ou ímpar?







Outro operador útil suportado pelo Scratch é o operador **round** (arredondamento de), que arredonda números decimais para o inteiro mais próximo. Por exemplo, **round(3.1) = 3**, **round(3.5) = 4** e **round(3.6) = 4**.

Números aleatórios

À medida que programar com mais frequência, é bem provável que você precisará gerar números aleatórios em algum momento, especialmente se você criar jogos e simulações. O Scratch disponibiliza o bloco **pick random** (escolha número entre) especificamente para esse propósito.

Esse bloco gera um número aleatório sempre que é utilizado. Suas duas caixas brancas editáveis permitem inserir um intervalo para esse número, e o Scratch selecionará valores somente entre esses dois limites (inclusive). A tabela 1.1 mostra alguns exemplos de uso desse bloco.

Tabela 1.1 – Exemplos de uso do bloco pick random (escolha número entre)

Exemplo	Resultados possíveis
	{0, 1}
	{0, 1, 2, 3, ... , 10}
	{-2, -1, 0, 1, 2}
	{0, 10, 20, 30, ... , 100}
	{0, 0.1, 0.15, 0.267, 0.3894, ... , 1.0}
	{0, 0.01, 0.12, 0.34, 0.58, ... , 1.0}

NOTA Os resultados de **pick random 0 to 1** (escolha número entre 0 e 1) e de **pick random 0 to 1.0** (escolha número entre 0 e 1.0) são diferentes. O primeiro caso resultará em 1 ou 0, porém o segundo fornecerá um valor decimal entre 0 e 1. Se alguma entrada do bloco **pick random** contiver um ponto decimal, a saída também será um valor decimal.

Funções matemáticas

O Scratch também suporta um grande número de funções matemáticas. O bloco **sqrt of** (raiz quadrada de) agrupa 14 funções matemáticas que podem ser selecionadas a partir do menu suspenso, incluindo raiz quadrada, funções trigonométricas, logarítmicas e exponenciais. Consulte *MathematicalFunctions.pdf* para obter uma descrição detalhada dessas funções.

Resumo

Este capítulo forneceu uma visão geral do Scratch e de seu ambiente de programação. Você conheceu os vários elementos da interface do usuário e até mesmo criou um jogo! Também exploramos os operadores e as funções matemáticas do Scratch.

A esta altura, você já conhece as informações básicas necessárias para criar alguns scripts eficazes no Scratch, porém esse é somente um passo na estrada para a criação de programas incríveis. Nos capítulos que se seguem, exploraremos mais profundamente o uso do Scratch para o desenvolvimento de suas habilidades de programação.

Problemas

1. Dê o resultado de cada bloco do script a seguir. Há um padrão nesses produtos?



2. Há um padrão nos produtos 9×9 , 99×99 , 999×999 , ... , etc.? Utilize o comando **say** (diga) para descobrir o resultado desses produtos e confira sua resposta.
3. Complete a tabela a seguir fornecendo o valor de cada expressão.

Expressão	Valor
$3 + (2 \times 5)$	
$(10 / 2) - 3$	
$7 + (8 \times 2) - 4$	
$(2 + 3) \times 4$	
$5 + (2 \times (7 - 4))$	
$(11 - 5) \times (2 + 1) / 2$	
$5 \times (5 + 4) - 2 \times (1 + 3)$	
$(6 + 12) \bmod 4$	
$3 \times (13 \bmod 3)$	
$5 + (17 \bmod 5) - 3$	

Agora utilize o comando **say** (diga) e os blocos apropriados de operadores para conferir suas respostas.

4. Avalie as expressões a seguir do Scratch usando lápis e papel. Suponha que $x = 2$ e $y = 4$.

a) 

d) 

b) 

e) 

c) 

5. Utilize o comando **say** (diga) e os blocos apropriados da paleta *Operators* (Operadores) para calcular o seguinte:
- A raiz quadrada de 32.
 - O seno de 30° .
 - O cosseno de 60° .
 - O resultado do arredondamento de 99,459.
6. Crie um bloco de função que calcule a média dos três números a seguir: 90, 95 e 98. Mostre o resultado usando o bloco **say** (diga).
7. Crie um bloco de função que converta 60°F para Celsius. (Dica: $C = (5/9) \times (F - 32)$.)
8. Crie um bloco de função que calcule a área de um trapezoide com altura igual a $4/6$ pés e bases iguais a $5/9$ pés e $22/9$ pés. (Dica: $A = 0,5 \times (b_1 + b_2) \times h$, em que h é a altura e b_1 e b_2 correspondem às duas bases.)
9. Crie um bloco de função que calcule a força necessária para acelerar um carro de 2.000 kg a 3 m/s^2 . (Dica: *Força = massa \times aceleração.*)
10. O custo da eletricidade é de 0,06 centavo de dólar por quilowatt-hora. Crie um bloco de função que calcule o custo de utilização de um ar-condicionado de 1.500 watts durante 2 horas. (Dica: *Energia = potência \times tempo*)
11. Com um simples truque matemático, você pode utilizar o operador **round** (arredondamento de) para arredondar um número para uma casa decimal específica. Por exemplo, você pode arredondar o número 5,3567 para o décimo mais próximo (ou seja, a primeira posição à direita da vírgula decimal) usando os três passos a seguir:
- $5,3567 \times 10 = 53,567$ (Multiplique o número por 10.)
 - $\text{round}(53,567) = 54$ (Arredonde a resposta do passo a.)
 - $54/10 = 5,4$ (Divida a resposta do passo b por 10.)

Que mudanças seriam necessárias nos passos anteriores para arredondar o número para o centésimo mais próximo (ou seja, para a segunda posição à direita da vírgula decimal)? Crie um bloco de função que arredonde 5,3567 para o décimo (ou centésimo) mais próximo e mostre o resultado usando o bloco **say** (diga).