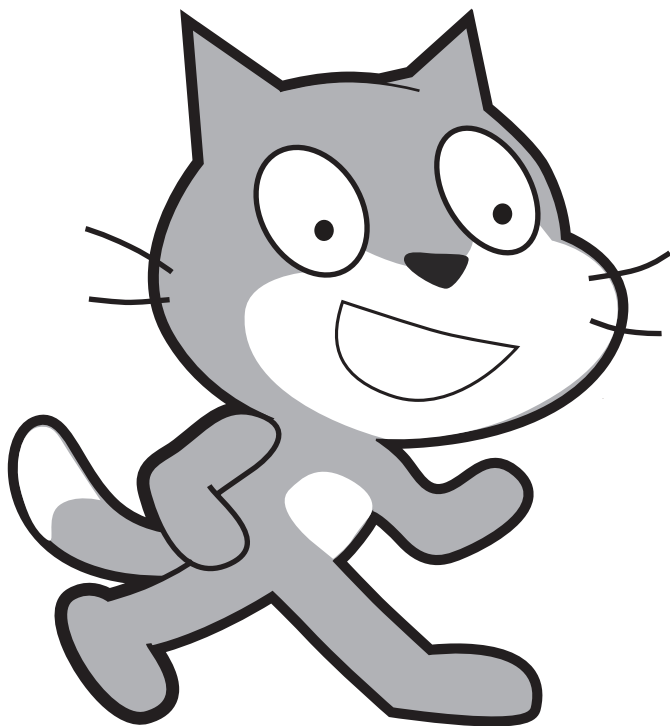


SCRATCH DAY

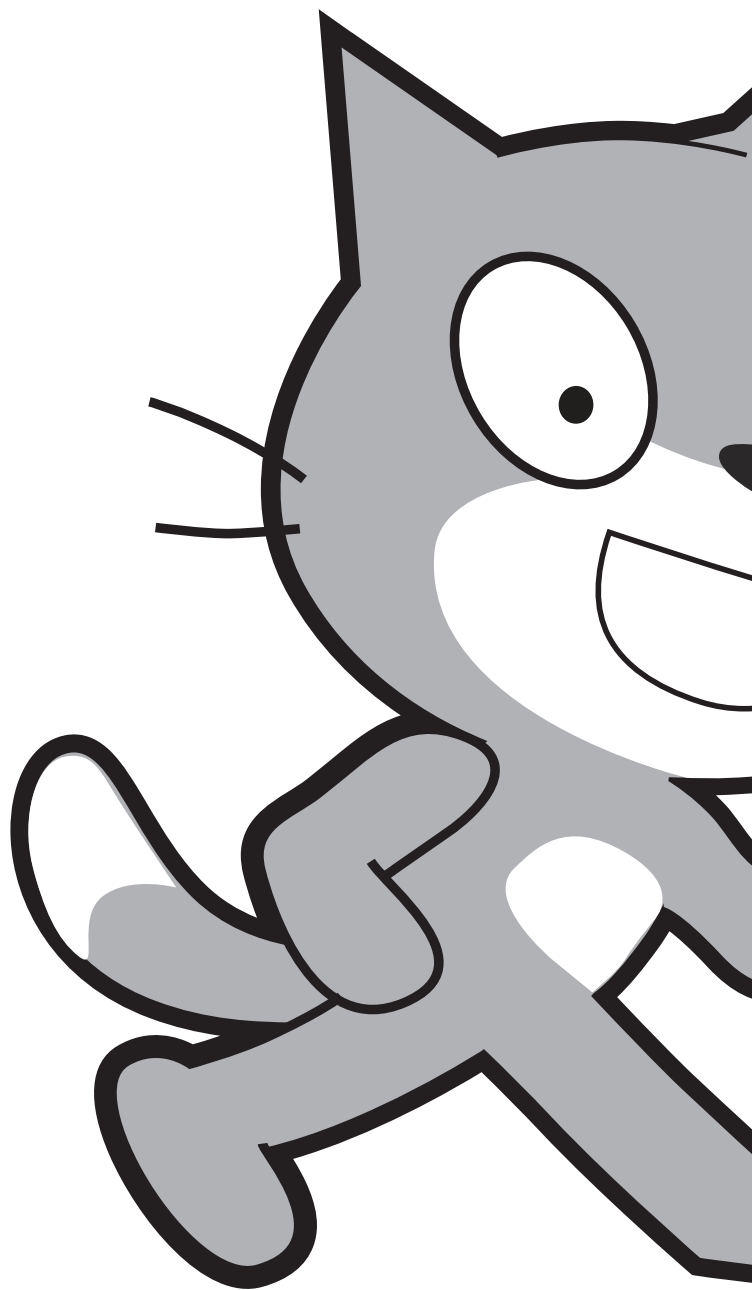
meet • share • learn

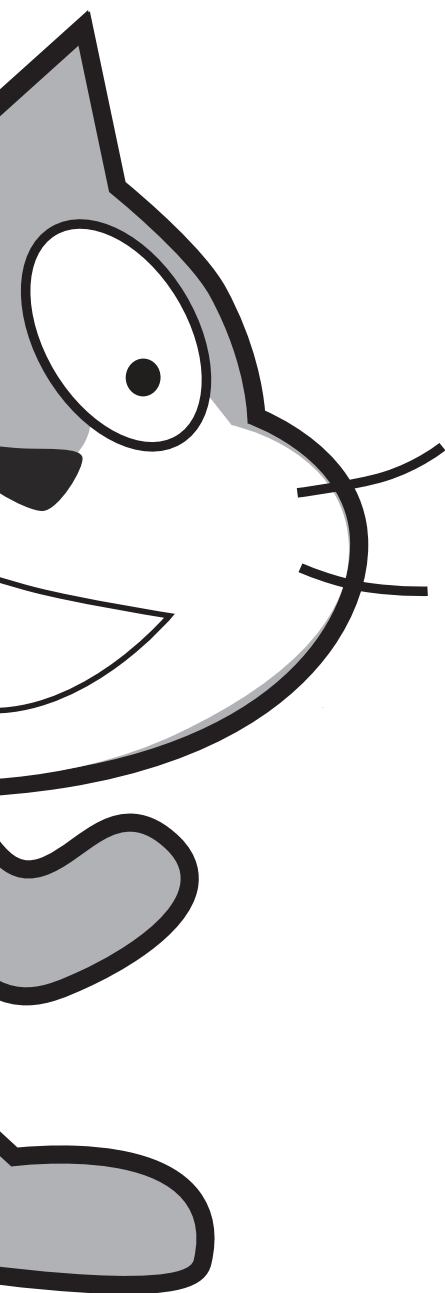


day.scratch.mit.edu



Laboratório e Observatório
de Ontologias Projetuais





Scratch Day

Cartilha v2.0

Bem-vindos, Scratchers!	4
Construcionismo	5
A interface	6
Programação com blocos	7
Blocos e objetos	8
Controle do programa	9
Laços e condicionais	10
Interatividade	11
Operadores e variáveis	12
Compartilhando projetos	13
Links úteis	14
Programas-exemplo	15

Bem-vindos, Scratchers!

Scratch Day é uma rede mundial de encontros nos quais pessoas se reúnem para encontrar outros Scratchers, compartilhar projetos e experiências, e aprender mais sobre a linguagem Scratch.

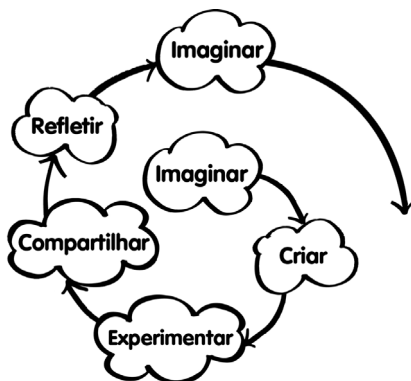
O principal objetivo da organização de um Scratch Day no Espírito Santo é difundir a abordagem construcionista para os processos de aprendizagem: acreditamos que a computação pode e deve ser aplicada para além da oferta de laboratórios de informática nas escolas. Os aprendizes podem experimentar oportunidades de utilizar a computação de maneira criativa, desafiadora e divertida, tendo a construção de projetos segundo seus próprios interesses como estratégia. O Scratch é uma importante ferramenta para essa abordagem e esperamos que possa ser útil.

Equipe responsável pelo material

Prof. Dr. Hugo Cristo Sant'Anna, André Victor da Silva Veronez, Fernando Gatti, Joyce Cavalcanti do Carmo, Larissa Firme Trabach, Mariane Azevedo Rocha, Paula Beltrão Zanotelli, Sara Rangel e Vinicius Bispo Neves.

A abordagem construcionista, elaborada por Seymour Papert no Massachusetts Institute of Technology a partir do construtivismo de Jean Piaget, defende que o aprendizado é um processo ativo onde as pessoas constroem o conhecimento com base nas suas experiências. Quando os aprendizes conseguem se engajar no processo e construir experiências significativas para eles, o aprendizado efetivamente ocorre.

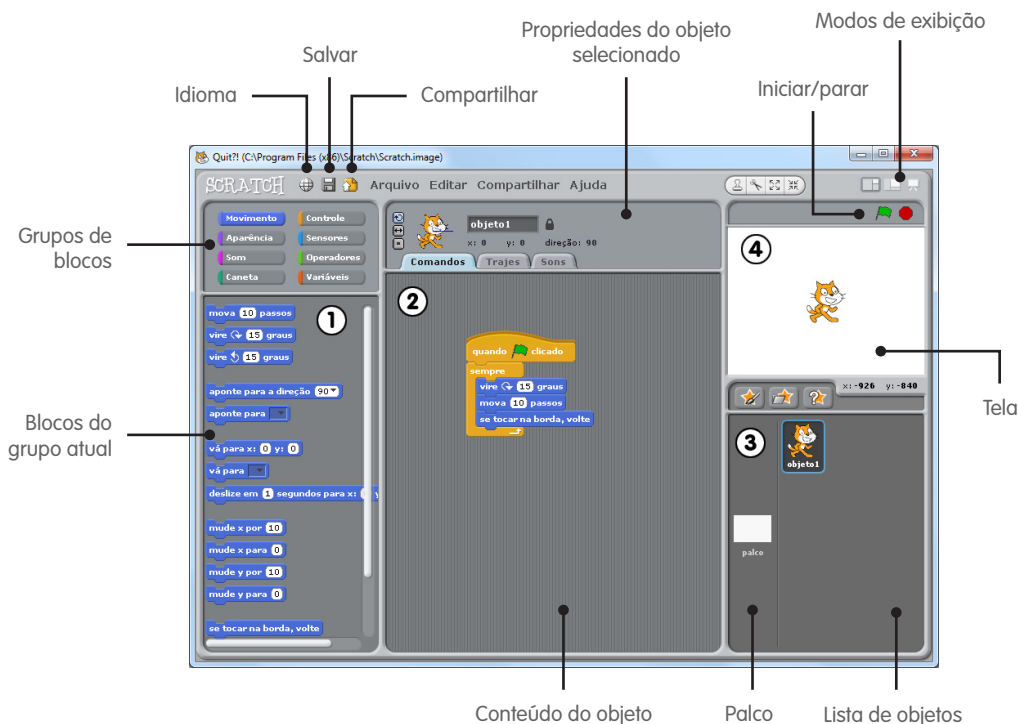
A espiral criativa do construcionismo representa um percurso onde os aprendizes imaginam o que desejam construir, criam projetos com base nessas idéias, experimentam alternativas, compartilham suas criações com seus pares e refletem sobre essas experiências. O Scratch é uma linguagem que realiza e dá suporte a esse percurso, transformando o computador em um ambiente propício ao desenvolvimento de projetos criativos apoiados pela Computação.



Para saber mais

O Lifelong Kindergarten do Massachusetts Institute of Technology disponibiliza uma série de publicações sobre os princípios do construcionismo no endereço <http://ilk.media.mit.edu/papers.php>.

A interface



A interface do Scratch é dividida em quatro áreas principais:

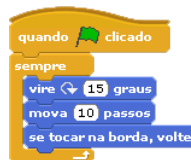
1. Os **Blocos** são os elementos dos programas, organizados em grupos de comandos com funções relacionadas.
2. As **Propriedades do objeto** listam o nome, posição X e Y, direção, controle de rotação do objeto e três abas que contêm: 1) comandos (programação com blocos); 2) trajes (aparência); 3) sons (efeitos sonoros).
3. A **Lista de objetos** contém todos os objetos que podem fazer parte do seu programa, incluindo o plano de fundo (o "palco").
4. A **Tela** é espaço para exibição e interação com os programas.

Programação com blocos

7

Programar em Scratch é muito parecido com montar coisas com LEGO. Os comandos são como blocos de construção que podem ser montados uns sobre os outros desde que o encaixe permita.

Os blocos do Scratch estão organizados por **cor** e por **forma**. A cor corresponde a que grupo o bloco pertence e a forma indica com quais outros blocos aquele pode ser encaixado. Há oito grupos de blocos no Scratch:



1. Movimento (azul): posicionam, movem e giram os objetos.
2. Aparência (roxo): mudam o traje, a cor, ampliam, reduzem, fazem aparecer/desaparecer, aplicam efeitos especiais e fazem os objetos "dizerem" coisas.
3. Som (vinho): reproduzem sons, tocam notas musicais e ritmos.
4. Caneta (verde): controlam a caneta ou o carimbo do objeto, utilizados para desenhar na tela do Scratch.
5. Controle (laranja): representam a lógica do programa - laços, condicionais, interações com teclado, mouse e demais objetos.
6. Sensores (azul claro): captam a situação de diversos elementos no Scratch - posição do mouse e dos objetos, digitação no teclado, colisão entre objetos e outros sensores especiais como LEGO Mindstorms ou controladoras Arduino.
7. Operadores (verde claro): fazem operações lógicas e matemáticas, utilizados junto com os blocos de controle.
8. Variáveis (abóbora): armazenam valores ou listas de valores.



Blocos e objetos

Os objetos no Scratch são como robôs aguardando por instruções sobre como se comportar nas mais diversas situações. Esse “comportamento” dos objetos resulta dos comandos escritos com combinações dos blocos. Os grupos de blocos da pág. 7 contêm os tipos de instruções que podemos enviar aos objetos.



Para enviar uma instrução para um objeto, basta clicar sobre ele na **Tela** ou na **Lista de objetos** (pág. 6), selecionar a aba **Comandos**, arrastar um bloco e soltar sobre a área **Conteúdo do objeto**.

O valor digitável nos blocos é chamado de **parâmetro** e pode ser modificado para um valor da sua preferência. Experimente mudar “10” no bloco **mude x por 10** para “20”. O efeito do bloco sobre o objeto atual pode ser experimentado com um clique sobre o bloco. Cliques sobre conjuntos de blocos encaixados resultarão na execução da sequência de blocos de cima para baixo.




Atividade

Experimente os blocos de movimento, aparência, som e caneta. Tente descobrir como cada um deles funciona e proponha combinações interessantes.

Controle do programa

9

Há duas principais funções para controlar os programas escritos na linguagem Scratch: **Iniciar comandos**  e **Parar tudo** .

Quando clicamos em  na interface, o Scratch executará em sequência (de cima para baixo) todos os comandos que estiverem encaixados no bloco correspondente. Enquanto o conjunto estiver sendo executado, haverá um contorno branco nos blocos.




Outras formas de executar comandos

Alguns programas requerem outros caminhos para iniciar a execução das sequências de comandos, especialmente quando aguardamos a interação do usuário em jogos, histórias multimídia e aplicativos. Para ajudar nesses casos, o Scratch conta com blocos de controle que permitem que uma sequência de comandos seja executada em situações específicas: quando o usuário clica em algum objeto, pressiona alguma tecla, ou quando os objetos trocam mensagens pré-definidas.



Atividade

Que tal modificar os programas da pág. 8 para serem iniciados quando clicamos em  ou utilizando os blocos acima? Você também pode experimentar os programas-exemplo da página 15 que utilizam esses blocos em situações um pouco diferentes.

Laços e condicionais

Alguns tipos de projetos contêm sequências de blocos que devem ser executadas mais de uma vez, como bolas que se movem em um jogo e notas que se repetem em uma música. Os **laços** são estruturas de controle que permitem a repetição no Scratch. Por exemplo, o bloco **repita** da direita combinado com o bloco **mova** apresenta o mesmo resultado dos três blocos **mova** da esquerda.



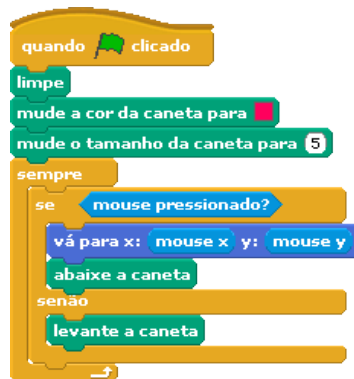
Além do bloco **repita**, o Scratch oferece outros que perpetuam a execução de um conjunto de blocos para sempre ou até que uma condição seja satisfeita: **sempre** e **repita até**. Esses blocos são especialmente úteis em jogos e animações que contêm programas que são executados muitas vezes.



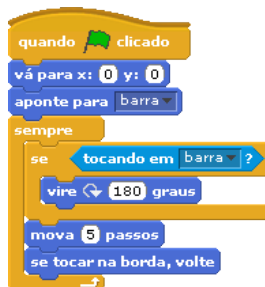
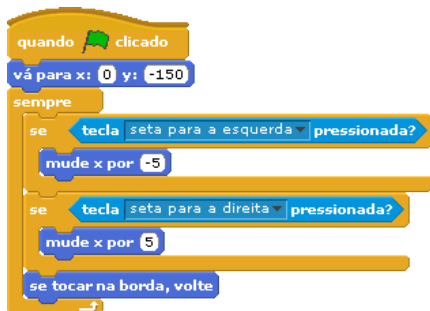
As condicionais são um outro tipo de estrutura de controle. Sua função é bloquear ou permitir a execução de um conjunto de blocos caso uma condição seja satisfeita (**se**) ou não (**senão**).



Resumidamente, a interatividade é projetada no Scratch por meio da combinação de sensores que capturam os dados enviados pelos usuários, via teclado e mouse, com laços e condicionais que processam e executam ações com base naqueles dados. O programa de desenho ao lado lê a posição do mouse e risca a tela caso um botão seja pressionado.



Os blocos a seguir implementam um jogo de paredão, também conhecido como Pong. Os blocos da direita controlam a bola (lembre-se dos robôs da pág.9) e os da esquerda controlam o jogador.



Atividade

Experimente adicionar bolas com velocidades diferentes ao jogo.

Operadores e variáveis

bola x**bola y**

O Pong da seção anterior é jogável, apesar de não ser muito divertido. A bola possui velocidade constante e após algumas jogadas a trajetória torna-se repetitiva. Uma forma de incrementar o movimento da bola seria utilizar variáveis para controlar seu comportamento: direção nos eixos X e Y, aceleração, rotação etc.

O Scratch permite a criação e monitoramento de variáveis, além de oferecer um conjunto completo de operadores lógicos e matemáticos (e, não, ou, adição, subtração, multiplicação, divisão, atribuição, comparação, maior que, menor que) além de funções trigonométricas, raiz quadrada, gerador de números aleatórios, logaritmos, arredondamento entre outras.



Os operadores são combinados aos blocos de condicionais e aos sensores do Scratch para ampliar as possibilidades interativas da linguagem, especialmente para jogos e aplicações multimídia.

Atividade

Tente desenvolver um conjunto de blocos que encerrará o jogo Pong da página 15 caso o jogador não consiga evitar que a bola toque na parte inferior da tela.

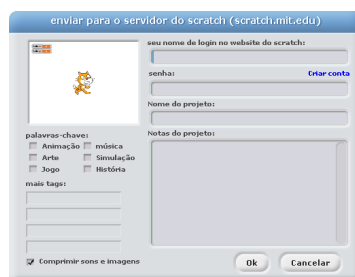
Compartilhando programas

13

A última etapa do ciclo de uso do Scratch é o compartilhamento dos projetos na comunidade on-line dos usuários da linguagem:



1. Acesse o site www.scratch.mit.edu e crie sua conta.
2. No Scratch, selecione o menu superior "Compartilhar" e a opção "compartilhar este projeto online..."
3. Preencha os campos descrevendo o seu projeto de forma que outros Scratchers possam encontrá-lo.



Para acessar o seu projeto, acesse a sua página no site do Scratch: www.scratch.mit.edu/users/seunomededeusuario.



Links úteis

Tutoriais e vídeos online

Vídeo-aulas em inglês e espanhol

<http://learnscratch.org/>

Material para professores

Scratch em Português

<http://kids.sapo.pt/scratch/>

Iniciando Scratch (PDF)

http://info.scratch.mit.edu/sites/infoscratch.media.mit.edu/files/file/GS_14-pt-BR.pdf

ScratchED: Recursos para educadores (em inglês)

<http://scratched.media.mit.edu/resources>

Scratch e sensores especiais

Kinect 2 Scratch - servidor do MS Kinect para o Scratch

http://scratch.saorog.com/?page_id=2

S4A - Scratch para a placa Arduino

<http://seaside.citilab.eu/scratch/arduino>

Controladora PicoCricket

<http://www.picocricket.com/>

Scratches alternativos

BYOB/Snap! - Scratch para Estudantes de Computação

<http://byob.berkeley.edu/>

A lista a seguir contém uma série de programas-exemplo que exploram diferentes questões no Scratch, do básico ao avançado. Recomendamos que você modifique cada um dos exemplos e tente compreender seu funcionamento.

Básico

<http://loop-ufes.org/scratchday/basico>

Ex1: Gato que anda, gira e volta ao tocar nas bordas (pág.7)

Ex2: Blocos e objetos (pág. 8)

Ex3: Blocos e objetos com controle (pág. 9)

Ex4: Laços e condicionais (pág.10)

Intermediário

<http://loop-ufes.org/scratchday/intermediario>

Ex5: Programa de desenho (pág. 11)

Ex6: Pong (pág.11)

Avançado

<http://loop-ufes.org/scratchday/avancado>

Ex7: Pong com variáveis (pág. 12)

Ex8: NicPong - Jabulani (Bônus!)



Vitória, ES - Brasil - 2015
www.loop-ufes.org