



Programação
KIDS

Programação Kids



Nome:



Sobre o curso

O nosso incrível curso Programação Kids é voltado exatamente para você! Que quer aprender os fundamentos para se tornar um desenvolvedor de respeito! Neste curso nós iremos ensinar a você de maneira simples e muito completa tudo o que é necessário para dominar a Lógica de Programação e o Pensamento Computacional sem usar telas pretas feias e códigos complicados que parecem não ter pé e nem cabeça.



Quantidade de Aulas
16 aulas



Carga horária
24 horas



Programas Utilizados
Scratch

O que aprender com este curso?

Ao longo desse curso, você desenvolverá muitas habilidades e mudará sua forma de ver receitas de bolos e programação.

Nós iremos aprender os conceitos básicos da programação, aprenderemos o que são algoritmos e rotinas, aprenderemos o que são variáveis e tipos primitivos, aprenderemos o que é o Scratch e como realizar saída de dados utilizando ele, aprenderemos também a realizar entrada de dados, e também a utilizar os operadores aritméticos na plataforma, aprenderemos o que são estruturas de controle, o que são operadores lógicos e como utilizá-los e claro, aprenderemos a utilizar tudo isso em estruturas de repetição. Vai ser um curso e tanto para você, feito com muito carinho e experiência!



Sumário

1 - O que é programação? É de comer?

- 1.1 - O que é a programação?
- 1.2 - A lista de instruções e as instruções.
- 1.3 - Exercícios

2 - Computadores (geralmente) nunca esquecem!

- 2.1 - O que são variáveis?
- 2.2 - Tipos de variáveis
 - 2.2.1 - Variável do tipo Texto
 - 2.2.2 - Variável do tipo Numérica
 - 2.2.3 - Variável do tipo Lógica
- 2.3 - Exercícios

3 - Scratch salva o dia dizendo "Olá, mundo!"

- 3.1 - Um manual do Scratch para você!
 - 3.1.1 - A guia do Código
 - 3.1.2 - A guia das Fantasias
 - 3.1.3 - A guia dos Sons
 - 3.1.4 - A Área de Trabalho
 - 3.1.5 - Um Palco para um Ator (Ou vários!)
- 3.2 - Olá, mundo! Eu estou aprendendo a programar!
- 3.3 - Exercícios

4 - Dino Sauro Rex da Silva

- 4.1 - Entrada de Dados
- 4.2 - O que são os Sensores?
 - 4.2.1 - Bloco de pergunta
 - 4.2.1.1 - Onde usar a variável resposta?
- 4.3 - Exercícios

5 - Não, isso não é uma referência!

- 5.1 - Entrada de dados
- 5.2 - Câmbio, alguém na escuta?
- 5.3 - Exercícios

6 - Scratch, o Matemático.

- 6.1 - Categoria Variáveis
- 6.2 - Categoria Operadores
- 6.3 - Exercícios

7 - O melhor jogo do mundo. O seu.

- 7.1 - Estruturas de Controle
- 7.2 - Exercícios

8 - Scratch, o gato com mais fôlego do mundo!

- 8.1 - Cenários
- 8.2 - Categoria Controle e Sensores

- 8.3 - Salvar e carregar projeto

- 8.4 - Exercícios

9 - Isso é maior, menor, ou a mesma coisa?

- 9.1 - Operadores Relacionais

- 9.2 - Exercícios

10 - Se o sorvete for de chocolate E a calda de morango, ele é perfeito!

- 10.1 - Operadores Lógicos

- 10.2 - Bloco "pare"

- 10.3 - Exercícios

11 - Mas se só um deles for de chocolate OU de morango, não tem problema!

- 11.1 - Operador Lógico "OU"

- 11.2 - Exercícios

12 - Gatos gostam de peixes, E também, de pontos!

- 12.0.1 - Blocos "quando ir for clicado" e "espere"

- 12.0.2 - Blocos "mostre", "número aleatório" e "vá para x: y:"

- 12.0.3 - Blocos "deslize", "posição x" e "esconda"

- 12.0.4 - Blocos "quando eu receber" e "se " então"

- 12.0.5 - Blocos "ou" e "menor que"

- 12.1 - Exercícios

13 - Haviam dois gatos, Peti e Repeti.

- 13.1 - Estruturas de Repetição

- 13.2 - Repita 10 vezes

- 13.3 - Sempre

- 13.4 - Repita até que

- 13.5 - Exercícios

14 - Olhe! Está chovendo peixes!

- 14.1 - Bloco "espere até que"

- 14.1.1 - Como usar o bloco "espere até que"

- 14.2 - Exercício

- 14.2.0.1 - Conjunto 1

- 14.2.0.2 - Conjunto 2

- 14.2.0.3 - Conjunto 3

15 - Aprimoramentos, gatos. Aprimoramentos.

- 15.1 - A variável "fantasia 'nome'/'número'"

- 15.2 - A biblioteca de sons do Scratch

- 15.3 - Exercícios

16 - Pode parecer o fim, mas é só o início.

- 16.1 - Exercício



Vamos começar!

Olá! Seja bem-vindo(a) ao primeiro capítulo da sua apostila! Aqui nós teremos informações complementares a sua aula! E adivinha quem é que te guiará pela apostila? Eu o Scratch!



1.1. O que é a programação?

Assim como eu citei na aula, a programação é basicamente quando a gente dá ordens ou instruções a algo. Simples assim! Mas nós ainda podemos ir muito longe nessa conversa, porque quando nós damos instruções, nós damos instruções a alguma coisa, como por exemplo:

Computadores, robôs, aviões, carros, seu celular, micro-ondas, máquinas de lavar, fones de ouvido, e acredite, até mesmo lâmpadas.

Quase tudo que conduz eletricidade pode receber microcomputadores, estes que podem receber essas instruções de que eu estou falando.

1.2. A lista de instruções e as instruções.

Uma receita de bolo, por assim dizer, pode receber um nome alternativo ao de "receita", esse nome é "Algoritmo". Muito sofisticado, né? Mas fica tranquilo, o que esse nome tem de bonito, ele tem de simples!

Algoritmos são basicamente listas ou receitas, nós usamos eles para fazer os computadores, fazerem coisas para a gente.

Agora as instruções das nossas receitas também recebem um nome bem sofisticado, esse nome é "Comando".

Os comandos formam os algoritmos, lembre-se disso, pois isso será útil para a sua vida toda! Agora vamos ver um exemplo.

Aqui temos uma receita de um certo hambúrguer de siri que consegui em uma das minhas aventuras.

O melhor hambúrguer de siri do mundo.

Ingredientes:

- Pão de 180 grãos, assado nas colinas úmidas das planícies do deserto de Nascar.
- Hambúrguer de carne de Siri Rei capturado na Fossa das Marianas, e assado por lá mesmo.
- Queijo feito em El Dorado.
- Alface e tomate orgânicos.
- Ketchup e maionese a gosto.

Método de preparo:

- Pegue o pão, e bote os condimentos.
- Depois coloque a alface e o tomate de maneira que a sombra deles se alinhe perfeitamente com o eclipse solar que ocorrerá no dia 29 de novembro no ano de 2049.
- Depois coloque o hambúrguer.
- Coloque o queijo.
- Feche o hambúrguer usando uma bigorna de ouro.



Essa é a receita, um pouco difícil de ser feita, mas o suficiente para entendermos.

Assim como você pode ver, todos os itens dentro da área "Modo de preparo", são comandos.

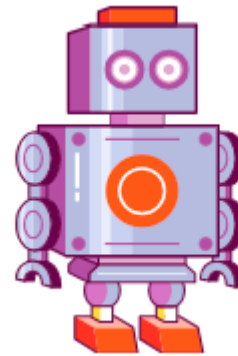
Agora, uma coisa bem importante deve ser dita: as máquinas, ou melhor dizendo, os computadores, precisam receber essas ordens de maneira muito clara.

Computadores não pensam como a gente, para ensinarmos um computador a fazer o nosso hambúrguer de siri perfeito, ele deve receber o máximo possível de instruções, e de maneira mais detalhada possível.

Por exemplo, se pedirmos para ele "Pegar o pão e botar os condimentos", assim como na receita, certamente ele exibirá uma mensagem de erro! Ou pior, vai fazer a receita de forma errada!

Nós precisamos dizer "Pegue o pão, e bote os condimentos no pão", aí, sim, teremos um

comando válido. Entendeu?



Agora, imagine que você está brincando com um robô de controle remoto. Esse robô é muito inteligente, mas só faz o que você diz a ele para fazer.

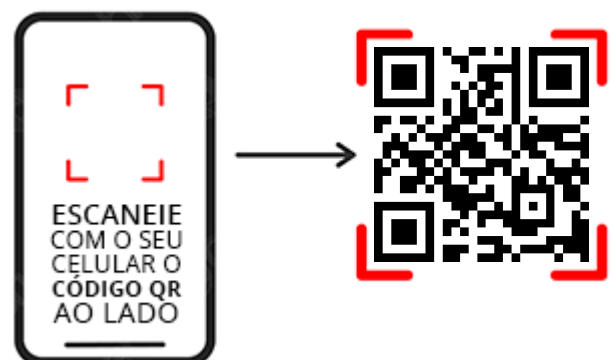
Para fazê-lo se mover para frente, você precisa apertar um botão específico no controle remoto. Para fazê-lo virar para a esquerda ou para a direita, você precisa apertar outros botões diferentes.

Agora, se você quiser que o robô faça uma dança divertida, você terá que apertar uma sequência de botões na ordem certa.

Agora, se anotarmos essa lista de movimentos, e colocarmos na memória do robô, estaremos programando ele para dançar!

É como ser o diretor de uma peça de teatro, dando todas as instruções para os atores, só que os atores são os programas de computador e o palco é a tela do computador ou do tablet!

Veja o exemplo que damos, na prática! Com um robô de verdade dançando!



<https://aposti.la/j8aj3>

1.3. Exercícios

No exercício dessa aula, nós iremos ensinar você a criar um tabuleiro para praticar a lógica de programação, assim como você viu na aula!

Ah, é importante você seguir os passos certinho, não vá fazer como o meu primo Luigi que esqueceu de botar combustível no avião! hehe.

Materiais:

Recomendações: faça acompanhado de seus pais ou responsável.

- Duas folhas de papel A4 ou uma cartolina e uma folha de papel A4.



- Lápis ou caneta.



- Régua.



- Tesoura sem ponta.



Como fazer:

- Usando um lápis ou uma caneta, desenhe quadrados de 5cm por 5cm na sua folha A4, ou na sua cartolina.

- Já, na outra folha A4, você desenhará quadrados de 4cm por 4cm. Ao todo devem ser 25 quadrados desenhados.

- Com a ajuda de um **adulto**, recorte-os.

- Após isso, desenhe em cada quadrado uma flecha.

- E logo após isso, do outro lado de cada quadrado, desenhe uma árvore, uma pedra ou um círculo. Ah, também deixe dois quadrados reservados! Em um deles você desenha uma bandeirinha e no outro um tesouro, uma estrela ou só escreva "fim", estes serão nossos pontos de início e fim do jogo.

- Feito! Agora é só colorir se desejar.

Você também pode usar EVA para fazer o tabuleiro e as pecinhas, fica muito legal!

Agora vamos ver como montar!

Primeiro passo: Posicione o tabuleiro e as peças dessa forma:



Olá, seja bem-vindo(a) ao segundo capítulo da sua apostila! Aqui veremos diferentes exemplos complementares a sua aula! Vamos seguindo adiante!



2.1. O que são variáveis?

Imagina que uma variável é como uma mala que guarda coisas diferentes. Essas coisas podem ser números, letras, cores, ou qualquer outra coisa que a gente queira lembrar. E o legal é que a gente pode dar um nome para essa mala!

Por exemplo, se a gente quer lembrar quantos doces temos, a gente pode chamar a nossa mala de “quantidade de doces” e colocar o número de doces lá dentro. E se a gente ganhar mais doces, a gente pode abrir a mala, mudar o número e guardar de novo.



Então, uma variável na programação é como essa mala que usamos para guardar e lembrar informações enquanto estamos fazendo coisas no computador!

2.2. Tipos de variáveis

2.2.1. Variável do tipo Texto

Imagine que você tem uma caixinha onde pode guardar palavras ou até mesmo uma historinha. Essa caixinha é como uma variável do tipo texto. Você pode colocar o nome de uma cor, o título de um livro, uma mensagem de aniversário ou até mesmo uma lista de compras dentro dela. Aliás, se juntarmos duas dessas variáveis, juntamos palavras!

Como, por exemplo, imagine uma variável de texto "Nome", e dentro dela está salvo "Ana".

Agora imagine a variável de texto "Sobrenome", e dentro dela está salvo a variável "da Silva".

Se juntarmos as duas, nós teremos "Ana da Silva!" Que legal, né?

2.2.2. Variável do tipo Numérica

São semelhantes às variáveis de Texto, porém, aqui apenas são guardados números! Então podemos guardar números como a sua idade, a quantidade de brinquedos que você tem, ou até mesmo o resultado de uma conta de matemática.

2.2.3. Variável do tipo Lógica

Estas variáveis são um pouco diferentes. Em vez de guardar números ou palavras, elas guardam apenas duas opções: verdadeiro ou falso. Por exemplo, se queremos saber se está chovendo lá fora, e olharmos pela janela, e estiver

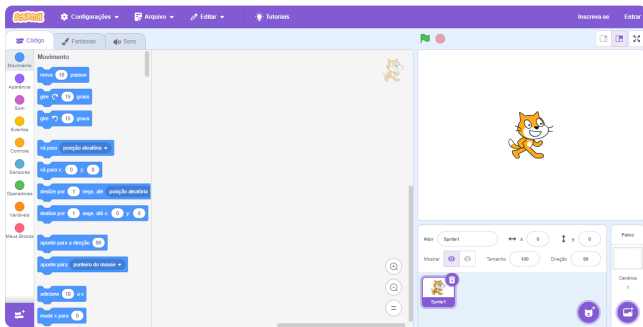


S seja bem-vindo(a) à mais um capítulo da nossa apostila! Neste capítulo nós veremos como é organizado e como funciona o Scratch!

E também, de quebra, veremos como criar nosso primeiro algoritmo no Scratch!

3.1. Um manual do Scratch para você!

Assim como nós vimos na nossa aula, esta é a tela inicial do Scratch! Veja bem e tente memorizar os botões e suas posições!



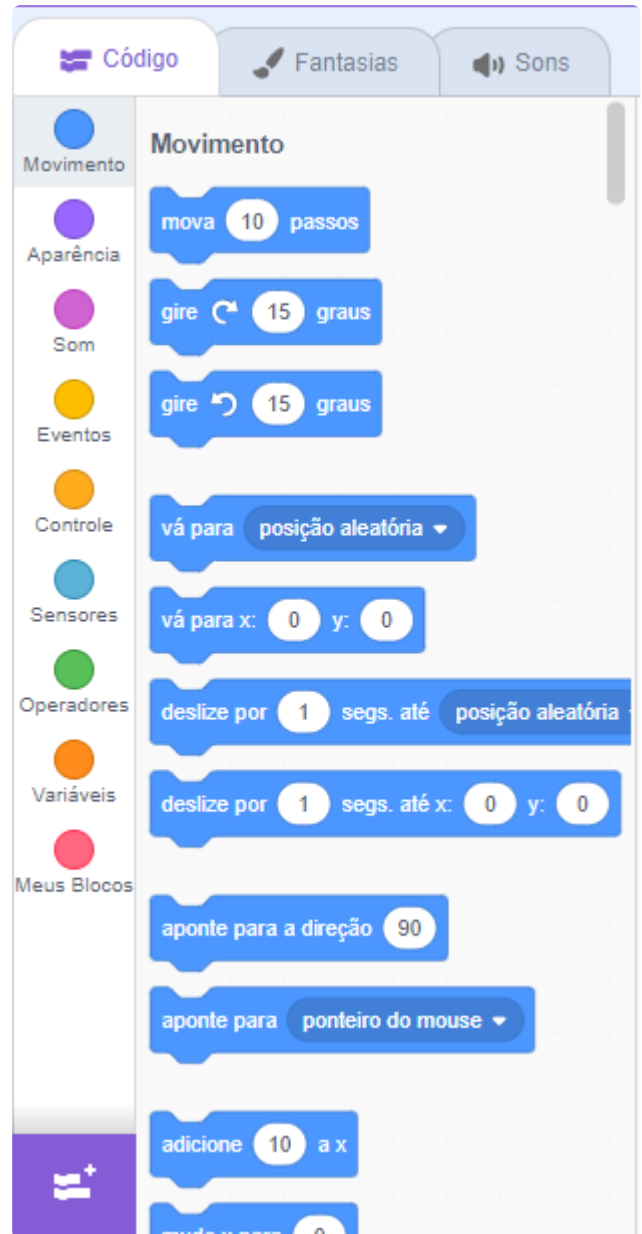
Você pode ver que assim como falamos na aula, o Scratch pode ser dividido em três partes. A primeira parte é a da esquerda! Aqui nós temos três guias!

3.1.1. A guia do Código

Temos a guia do Código, com as suas categorias de bloco de comando, operadores e variáveis. Estes serão abordados individualmente nas aulas seguintes!

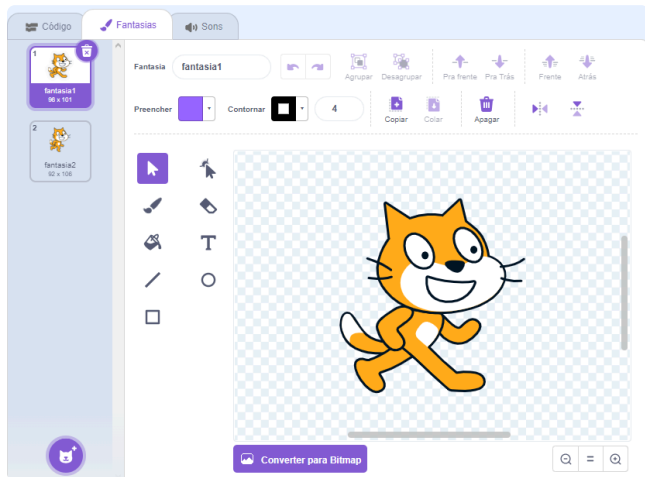
Mas bem, tudo o que precisamos saber e lembrar é que os blocos de comandos no Scratch se ligam como peças de um quebra-cabeça, mas nem de longe são difíceis como um! Aliás, é importante dizer que os blocos de comandos funcionam de cima para baixo e sempre da esquerda para a direita. :)

Veja só como o campo é em detalhes!



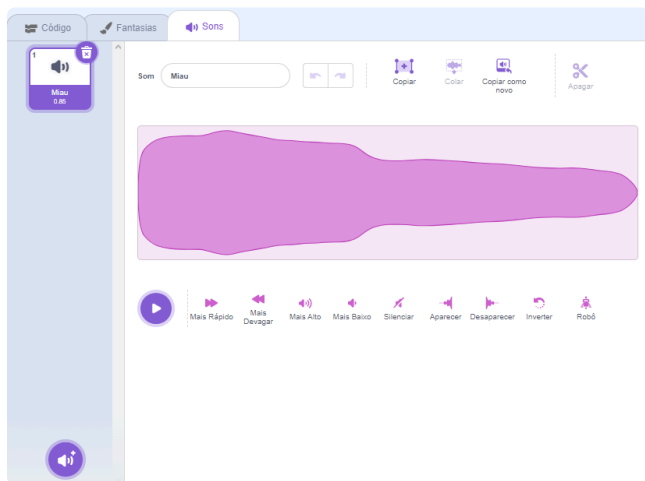
3.1.2. A guia das Fantasia

Aqui nós podemos editar a minha bela imagem! Ou ainda, se você quiser, pode editar aqui a fantasia de outros atores e cenários! Mas você tem que selecionar o ator ou o cenário antes de fazer isso, ok?



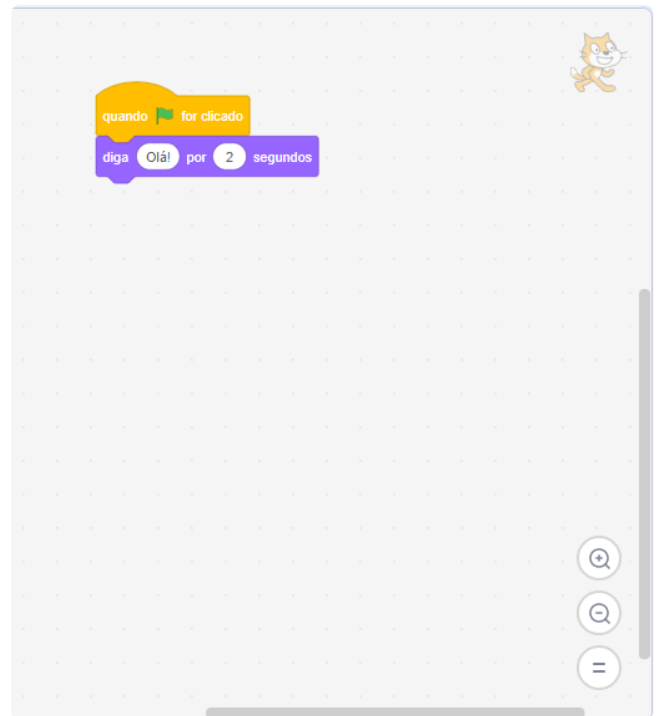
3.1.3. A guia dos Sons

Aqui, similar à guia das Fantasia, nós temos como editar os sons! Podendo adicionar sons conforme a nossa necessidade, mutar, deixar mais rápido ou devagar, mais alto ou mais baixo entre outras coisas muito maneiras!

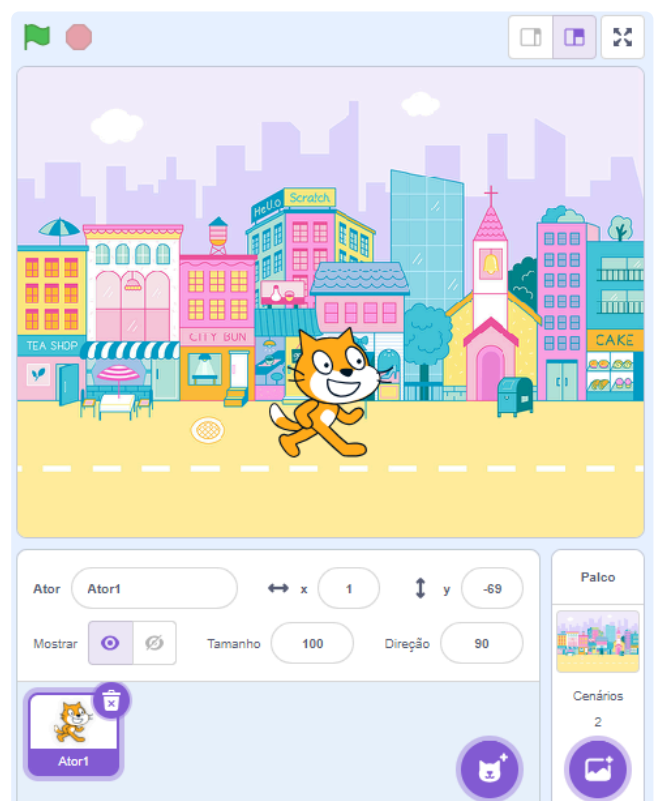


3.1.4. A Área de Trabalho

E aqui no centro da tela, temos a Área e Trabalho, que é o lugar onde nós iremos colocar nossos blocos de comandos! Basta clicar no bloco que você quer, manter o clique e arrastar até o lugar que você deseja na Área de Trabalho!



3.1.5. Um Palco para um Ator (Ou vários!)



3.2. Olá, mundo! Eu estou aprendendo a programar!

Agora que vimos onde achamos os nossos tão queridos blocos e como e onde colocamos eles, estamos prontos para seguir!

Veja o seguinte bloco:



Você sabe de qual categoria ele veio? O bloco "Diga" faz o nosso ator dizer "Olá!" por 2 segundos, aliás assim como vimos na aula, para fazer um comando funcionar basta clicar nele, ou quando usamos um bloco de comando da categoria "Eventos", que veremos em seguida.

Ah, e ele veio da categoria "Aparência"!

Agora você se lembra dos exemplos que demos na Aula 1? As setas? Esses blocos são como elas e eles fazem o nosso Ator realizar os comandos contidos neles. Legal, né?

Agora vamos ver o seguinte bloco:



E este bloco, você sabe de qual categoria ele veio? Essa é fácil! Da categoria "Eventos"!

Os blocos dessa categoria instruem o nosso computador a realizar uma ação QUANDO alguma coisa acontecer, é bem simples, mas entraremos em detalhes em aulas futuras!

3.3. Exercícios

No primeiro exercício seu objetivo será bem simples!

Digite o seguinte link no seu navegador: <https://aposti.la/pmqh7>

Depois disso, aperte o "Enter" ou clique na "lupinha" para pesquisar.

E se divirta! O seu objetivo será abrir o Scratch e testar os blocos de comando! Não tenha medo de mexer e brincar com os blocos! Junte eles, separe-os, teste diferentes combinações, mexa e remexa! Mas também tente encontrar os seguintes blocos de comandos:

1. "adicione '10' a X".
2. "esconda".
3. "volume".
4. "repita até que".
5. "resposta".
6. "nome de usuário".
7. "não".
8. "my variable".

Já o seu segundo exercício será fazer o nosso ator dizer outras 3 frases!

Elas podem ser qualquer coisa, como, por exemplo, "Eu sou um gato laranja!", ou "Eu gosto de brócolis!", fica a seu gosto! Já que eu gosto mesmo de brócolis.

Não esqueça que, para fazer os comandos funcionarem, é só clicar neles!

Ah, também dá para fazer isso usando os blocos de eventos! Mas não se preocupe caso não consiga usar esses blocos, nós iremos explicar em detalhes depois.



S seja bem-vindo(a) à mais um capítulo da apostila, nesse capítulo teremos uma explicação mais detalhada sobre o conteúdo da aula!

4.1. Entrada de Dados

As entradas de dados são informações que entram em um sistema de computador. Essas informações podem ser fornecidas de várias maneiras, como por meio de um teclado, mouse, microfone, tela sensível ao toque ou qualquer outro dispositivo.

As entradas de dados são essenciais para que os computadores possam receber instruções, processar informações e realizar tarefas específicas de acordo com as solicitações dos usuários.

Por exemplo, quando você digita texto em um documento, clica em um botão em um site da web ou fala com um assistente de voz, você está fornecendo entradas de dados para o computador.

Em resumo, as entradas de dados são a forma como os usuários interagem e comunicam com os sistemas de computador.

4.2. O que são os Sensores?

No Scratch, os sensores são como “sentidos” que permitem que o seu programa saiba o que está acontecendo ao redor. Eles são responsáveis por fazer o seu projeto reagir a diferentes condições do ambiente.

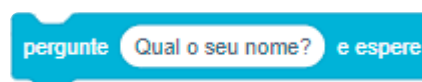
Por exemplo, um sensor pode ser um detector de colisões, que percebe quando um objeto toca em outro no jogo que você criou. Outro exemplo é o sensor de tecla pressionada,

que percebe quando você pressiona uma tecla no teclado.

Esses sensores ajudam o seu programa a tomar decisões com base no que está acontecendo no jogo ou na animação que você fez. Então, você pode programar o seu projeto para fazer algo específico quando um sensor detectar alguma coisa.

4.2.1. Bloco de pergunta

O bloco “Pergunta” é uma ferramenta no Scratch que permite que você faça perguntas aos usuários e receba suas respostas dentro do seu projeto.



Quando você usa o bloco “Pergunta”, o Scratch exibe uma caixa de diálogo na tela do usuário, onde ele pode digitar uma resposta.

Por exemplo, você pode criar um jogo onde o personagem pergunta ao jogador “Qual é o seu nome?”. Quando o bloco “Pergunta” é executado durante o jogo, uma caixa de diálogo aparece na tela, permitindo que o jogador digite seu nome como resposta.

Depois que o jogador digita sua resposta e pressiona “Enter” ou “OK”, o Scratch armazena essa resposta em uma variável. Você pode então usar essa variável para fazer com que seu projeto responda de maneira personalizada com base na resposta do usuário.

4.2.1.1. Onde usar a variável resposta?

Como foi explicado em aula, o bloco “resposta” armazena a última resposta digitada no bloco de pergunta, mas esse bloco tem um probleminha.

Zipt-zapt-zuuuuuum! Que o gato laranja fique grandauum!

Como é bom te ver aqui meu brotinho, vejo que suas raízes continuam a crescer!

Neste capítulo vamos continuar falando de entrada de dados!

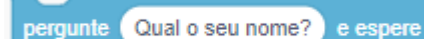


5.1. Entrada de dados

E não, não é para pensar em uma porta onde apenas dados daqueles de tabuleiro passam, no mundo da computação, dados são todos os tipos de informações! Sabe, tudo são dados, desde o valor da marmitta no mercadinho, até a fofoca que nossos amigos fazem quando ficam sabendo de um babado.



E não é difícil trabalhar com essas coisas! No Scratch o bloco "pergunte 'Qual o seu nome'" faz isso bem fácil. Essa é uma das formas que temos de nos comunicarmos e interagirmos com o nosso código, e com o computador.



pergunte Qual o seu nome? e espere

Mas esse bloco não aceita apenas entradas de texto, o Scratch é um aplicativo muito inteligente, ele consegue reconhecer números e os diferenciar de textos!

Com no exemplo da aula 5, nós temos usado o mesmo bloco, mas pedindo um número e ele reconheceu sem problemas! Mas agora, talvez, você deva estar se perguntado: o que acontece se eu digitar algum texto? A resposta é simples, nada!

5.2. Câmbio, alguém na escuta?

Você se lembra o que foi de diferente que fizemos na aula 5? Nós fizemos dois atores interagirem! Mas isso a princípio não seria impossível? Afinal estes dois possuem códigos, algoritmos diferentes, certo?

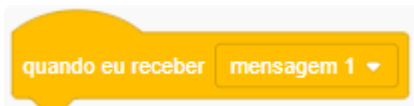
Bem meu brotinho, a resposta é que sim, os códigos são separados e não, não é impossível.

Afinal, estamos ainda no mesmo projeto! Ou seja, é tudo conectado!

Com isso nós só precisaríamos de um comando, que deixasse o outro ator sabendo que alguma coisa aconteceu, né? Tipo como quando nós já temos a resposta da pergunta "Quanto eu tenho que fazer o Scratch crescer?". Bem, nós temos esse comando, e o nome dele é "transmita 'mensagem 1' e espere", e este comando fica na guia "Eventos".



Esse comando envia uma "mensagem" para todos os que puderem ouvir, a mensagem em si não importa, o que importa é que ela foi enviada, e que cada mensagem é diferente, assim podemos detectar quando ela é enviada usando outro comando, também da guia "Eventos". Me refiro ao comando "quando eu receber 'mensagem 1'".

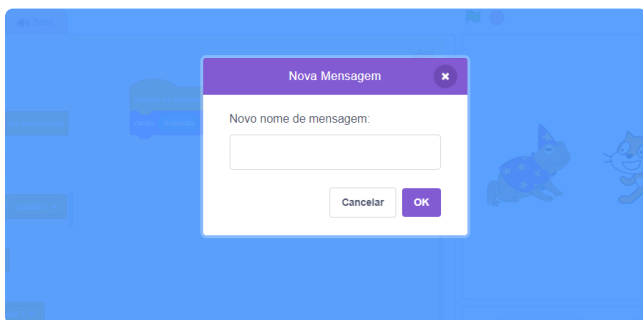


E qualquer ator que possuir esse bloco em seu código será capaz de "receber" essa mensagem.

Ah, o nome da mensagem pode mudar, basta você primeiro clicar na caixinha do comando "transmita 'mensagem 1'".



Em seguida clicamos em "Nova mensagem".



Aí é só digitar o nome dessa nossa nova mensagem, ou seria a mensagem em si? Seja criativo! Hehe.

Ah, e por último ainda temos o comando "mude '10' no tamanho", que aumenta o tamanho total do nosso ator, ele fica na guia "Aparência".

O mais legal desse comando, é que se digitarmos um número positivo, ou seja, aqueles que vem depois do 0, o Scratch irá crescer, mas se por acaso digitarmos um número negativo, como: -1, -2, -3... O Scratch irá encolher!



5.3. Exercícios

No exercício da aula de hoje, nosso desafio será bem interessante, pois o Sapo Mago sabe fazer mais uma magia! E nós vamos ensinar você a fazer essa magia usando o Scratch! A magia será fazer o Sapo Mago mover tanto para cima ou para baixo, ou para direita ou para a esquerda, o nosso ator! Euzinho, o Scratch!

1. Vamos lá, agora nós iremos escolher dois atores, podem ser os que você quiser, para escolher eles basta clicar no botão "Selecione um ator" e escolher os que você mais gostar.



2. Após escolher os atores e posicionar eles no palco, do jeito que você quiser, vamos selecionar o primeiro ator, clicando no ícone dele

3. Após isso, vamos selecionar a guia superior "Código".

4. Então vamos a guia lateral "Eventos".

5. Na sequência, vamos arrastar até o nosso código nosso primeiro bloco, o bloco "quando 'ir' for clicado".

6. Após isso vamos selecionar a guia "Sensores".

8. Vamos selecionar o bloco "pergunte 'qual o seu nome?' e espere", e vamos conectar ele abaixo do bloco "quando 'ir' for clicado".

9. Vamos mudar o conteúdo do bloco "pergunte 'qual o seu nome?' e espere", para isso apague o texto dele e digite "Quanto eu devo mover o Scratch para cima ou para baixo?".

10. Vamos clicar novamente em "Eventos".

11. Na sequência vamos clicar e arrastar o bloco "transmita 'mensagem 1' e espere" até conectarmos ele abaixo do bloco superior.

12. Vamos voltar para os "Sensores".

13. Conecte mais uma vez o bloco "pergunte 'Qual é o seu nome?' e espere" abaixo do último bloco posicionado.

14. Assim como você fez da última vez, mude o conteúdo do texto desse bloco para "Quanto eu devo mover o Scratch para esquerda ou para direita?".

15. Voltamos para a guia dos "Eventos".

16. Posicione abaixo do último bloco posicionado o bloco "transmita 'mensagem 1' e espere".

17. Temos que mudar a mensagem ou teremos problemas no nosso código, então vamos clicar na caixinha dentro do bloco.

18. Clique em "Nova mensagem".

19. Nomeie a nova mensagem como "mensagem 2", você também pode escolher outro nome, mas estamos usando este para facilitar o entendimento.

20. Vamos agora clicar no nosso segundo ator, e também na guia superior "Código", vamos escolher na guia lateral "Eventos" o bloco "Quando eu receber 'mensagem 1'" e posicionar ele no nosso código, que ao momento deve estar vazio.

21. Vamos agora na guia "Movimento".

22. Vamos conectar o bloco "adicione '10' a y" ao bloco anterior.

23. Volte agora para a guia "Eventos".

24. Mova novamente para o campo do código o bloco "quando eu receber 'mensagem 1'", mas clique na caixa e mude a "mensagem1" para "mensagem 2". Você verá que terá basicamente dois códigos no campo, mas isso não tem problema!

25. Então, volte uma última vez para a guia "Movimento", e conecte no bloco anterior o bloco "adicione '10' a x".

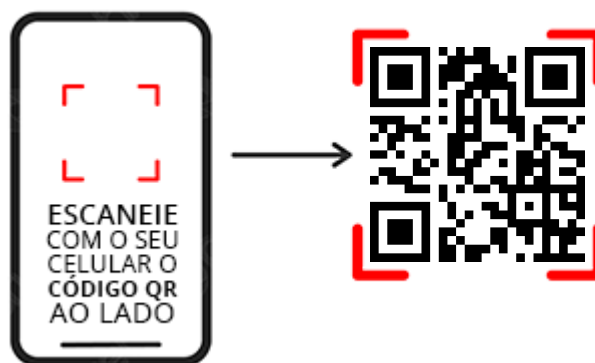
26. Então, vá para a guia "Sensores".

27. Aqui, você irá finalmente clicar e arrastar a variável "resposta" até as caixas dos dois blocos de "adicione '10' a y" e adicione '10' a x".

28. Então é só clicar no botão de "Ir" e se divertir! Vale destacar que você pode digitar tanto números negativos quanto positivos para mover o nosso segundo ator na direção que você quiser! Faça seus testes!



Aliás, você pode testar o código por aqui, é só acessar o QR Code abaixo ou copiar o link e colar ele na barra de pesquisas do seu navegador!



<https://aposti.la/he3n0>

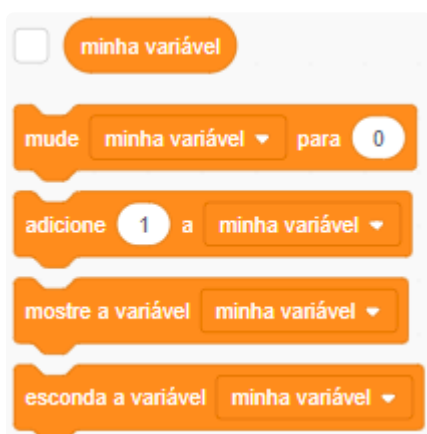


S seja bem-vindo(a) à mais um capítulo da nossa apostila! Neste capítulo irei lhe explicar o que são e para que servem as categorias mencionadas durante a aula, que eram a categoria de Operadores e a categoria de Variáveis.

6.1. Categoria Variáveis

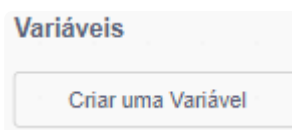
As variáveis do Scratch são bastante importantes, ainda mais se você deseja criar um algoritmo mais complexo, que você precise de variáveis, pois como já havia sido falado, as variáveis servem para você armazenar textos ou números!

Podemos definir o valor a ser guardado em uma variável de forma manual, utilizando o bloco “Adicione”, que está presente logo abaixo das variáveis existentes.



Também podemos definir o valor nessa variável através do bloco “resposta”, que se encontra na categoria de Sensores!

Então, nessas variáveis podemos armazenar textos e números, além de podermos criar uma variável específica para um ator, ou para todos.



Um exemplo muito legal de onde podemos utilizar uma variável é: imagine que você criou um jogo, e então você precisa que o jogador digite o nome dele, então você pode criar uma variável chamada “Nome do jogador” e que quando ele perder ou vencer no jogo, o nome dele irá aparecer na tela!



6.2. Categoria Operadores

A categoria dos Operadores é responsável por realizar operações matemáticas e operações lógicas. Eles incluem operadores matemáticos que vimos durante a aula, como o de adição (+), subtração (-), multiplicação (*) e divisão (/).



Em resumo, parte da categoria é responsável por manipular valores e expressões, realizar cálculos, comparar valores e tomar decisões com base em condições específicas.

Sendo fundamentais para a construção de algoritmos e lógica de programação no geral.

Chegamos agora ao 7º capítulo da nossa apostila, e aqui vamos falar sobre as estruturas de controle!

Vamos dar exemplos de como elas funcionam e no exercício, faremos uma versão mais interessante do código que montamos!



7.1. Estruturas de Controle

Vamos começar a falar sobre o principal bloco de comando da categoria "Controle" no Scratch, me refiro ao bloco "se então".



Como dito anteriormente na aula, esse bloco parece muito com uma grande pedra em um rio, dividindo o curso que a água percorre, mas diferente disso, nós controlamos exatamente o curso que a água segue.

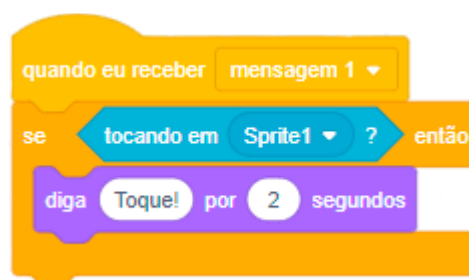
Veja, por exemplo, o hexágono que fica entre o "se" e o "então", esse espaço é reservado para colocarmos os famosos operadores relacionais, bem não se preocupe se você nunca ouviu falar deles, é que eles são que nem banda estrangeira, só quando se é fã, você percebe o quão famoso é!

E como o Scratch facilita as coisas para a gente, nós temos alguns blocos da categoria "Sensores" que também podemos usar nesse local, como o que vimos nessa aula, "tocando em 'ponteiro do mouse' ?".



E ainda, indo mais além, quando usamos esses blocos, nós estabelecemos uma "condição", que quando atendida, ativa o código que está dentro do nosso bloco.

Veja um exemplo de código que usamos nessa aula.



No exemplo acima nós temos duas condições que devem ser atendidas, a primeira é a "quando eu receber 'mensagem 1'", que diz, dessa forma, que o código abaixo dela só será executado quando o bloco "transmita 'mensagem 1'" no outro ator, ou atores, for executado.

E ainda, temos a estrutura "se então", nesse bloco, ou em qualquer outro que receba outros blocos em formato de colmeia, nós temos outra condição que deve ser atendida, ou não, para funcionar. Por exemplo, neste caso, mesmo que a

primeira condição seja verdadeira, ou no caso, o bloco realmente tenha recebido a 'mensagem 1', o nosso ator só irá dizer "Toque!" SE, ele estiver tocando no ator "Sprite1".

Meu brotinho, minha brotinha, não se preocupe caso isso tudo pareça difícil, acredite em mim, é bem simples!

Não tenha medo de ler ou reler a explicação, mesmo os maiores gênios da humanidade não nasceram sabendo! E todos temos o nosso tempo para aprender as coisas!

Às vezes podemos acreditar que não somos capazes de aprender algo, mas se você passa por isso, acredite em mim que eu acredito em você, com certeza você consegue fazer qualquer coisa no mundo! Com o devido estudo e preparo, é claro! Hehehe...

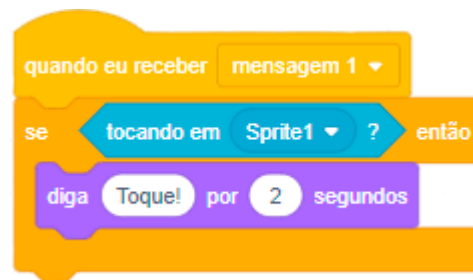
7.2. Exercícios

No exercício desse capítulo, nós vamos reutilizar o nosso código da Prática na Aula 7 para fazermos o Scratch "chutar" a bola que usamos de exemplo, de um lado para o outro, e para isso precisaremos fazer algumas alterações nesse nosso código.

Bem, para fins de controle, vamos precisar usar o código que escrevemos na aula passada, né? Veja eles logo abaixo, só para podermos começar sem problemas.



Código do ator "Sprite1".



Código do ator "Soccer Ball".

1. Certo, a primeira coisa que podemos fazer é mudar o nosso cenário, nós podemos fazer isso no botão "Selecionar Cenário".

2. E após isso, basta procurar pelo cenário "Soccer 2".

3. Então, após isso, voltando para o ator "Sprite1".

4. Clique na caixa do bloco "transmita 'mensagem 1'" do algoritmo da direita.

5. Após isso, clique em "Nova mensagem", e crie uma nova mensagem batizando ela de "mensagem 2", e após isso, selecione esta nesse bloco.

6. Então, após isso, volte no ator "Soccer Ball".

7. Clique no topo do algoritmo com o botão direito do mouse.

8. Clique em "Duplicar", e arraste o novo bloco de algoritmos até abaixo do anterior.

9. No novo algoritmo, clique na caixa de "quando eu receber 'mensagem 1'", e mude a mensagem do bloco para "mensagem 2".

10. Após isso, retire o bloco "diga 'Toque!' por '2' segundos" de dentro do bloco "se então", de ambos os algoritmos.

11. E após isso, indo na categoria "Movimento", clique e arraste o bloco "deslize por '1' segs. até x: y: ", até dentro dos blocos "se então", de ambos os algoritmos.

12. Então, clique em "Operadores".

13. Clique e arraste os blocos de "subtraia"



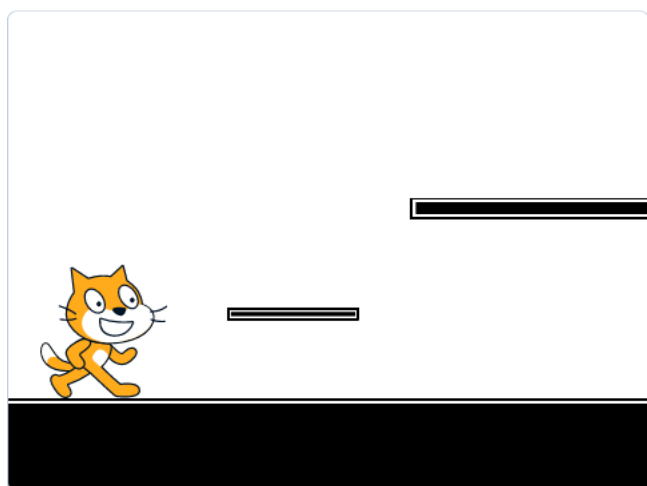
Olá, tudo bem meu brotinho? Seja bem-vindo(a) a mais um capítulo da nossa apostila, hoje nós vamos rever explicações sobre categorias e blocos que foram e voltarão a ser usados no futuro, ou seja, basicamente vamos rever as explicações do que vimos nessa aula, isso para fixação do conteúdo, e também vou lhe ensinar a salvar seus projetos feitos na plataforma!

8.1. Cenários

Os cenários têm muita importância na plataforma, imagina você usar a plataforma sem poder colocar cenários, o seu ator ficaria em um limbo! Uma caixa vazia quase sem fim!

Mas que bom que eles existem, e eles servem justamente para serem... cenários, ora!

Eles possuem efeitos únicos, como os de transição de cenário, ou seja, digamos que você queira criar um joguinho de plataforma, então, você mesmo pode criar suas plataformas, utilizando a própria ferramenta de criação que a plataforma disponibiliza, como o exemplo a seguir, além de também poder escolher outros cenários, que já estão nela.

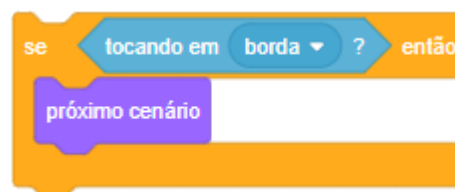


“Mas como eu posso fazer essa transição de cenários?”, é muito simples, fique ligado(a) no exercício dessa apostila, nele vou lhe explicar de uma forma simples, como fazer essa tal transição entre cenários!

8.2. Categoria Controle e Sensores

Então, como já utilizamos em aulas passadas os blocos dessas duas categorias e, que também já foram explicadas, hoje vou lhe mostrar onde você pode usá-los!

Um exemplo bem comum seria criar um código para verificar se o ator está encostando em algo, esse exemplo é bastante utilizado para criar joguinhos dentro da plataforma, e pode servir para fazer a própria troca de cenário que veremos no exercício ou até mesmo causar algum dano ao ator, caso ele toque em um inimigo, por exemplo, as possibilidades são infinitas!

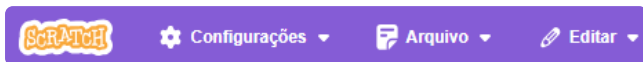


Note-se também que sempre tem um bloco de controle presente, pois ele é fundamental para fazer o código, sendo a base principal, ou seja, a partir dele que podemos fazer as verificações, seja com algum bloco de sensor, operador, etc.

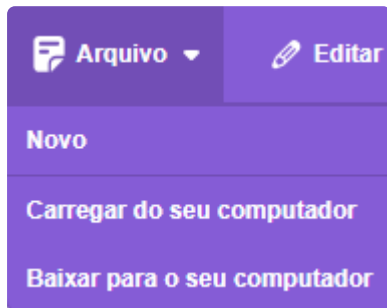
8.3. Salvar e carregar projeto

Agora, vamos a uma explicação rápida de como salvar o arquivo de um projeto da plataforma! Por enquanto apenas vou lhe ensinar a como salvar ele em formato de arquivo na sua máquina, quando avançarmos no conteúdo ensinaremos a salvar na própria plataforma!

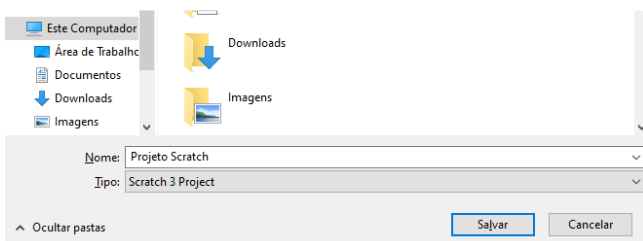
Então, para salvar em formato de arquivo você deve clicar no botão “Arquivo”, igual ao exemplo:



Após clicar em “Arquivo”, você deve clicar em “Baixar para o seu computador”.



E após clicar ali, você deve escolher o local para salvar seu projeto, agora é por sua conta, você pode escolher o lugar que você quiser para salvar.

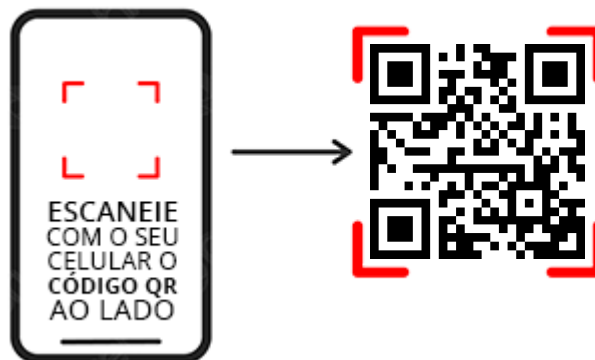


Então é só clicar em “Salvar”, que o arquivo será salvo como formato “SB3”, que é o formato de arquivo que a plataforma ou aplicativo do Scratch utiliza para leitura.

E para abrir o projeto não é tão diferente, apenas basta seguir os mesmos passos, porém ao invés de clicar em “baixar para o seu computador”, você deve clicar em “Carregar do seu computador”, deve ser um pouco complicado achar o arquivo caso você se esqueça de onde salvou, então, trate de salvar em um local que você utiliza bastante, como, por exemplo, a “área de trabalho”.

8.4. Exercícios

Então, vou lhe ensinar a fazer uma transição de cenários dentro da plataforma, abra o link a seguir na plataforma, ele já tem o código para fazer nosso ator andar, então basta apenas criar o código para transição, confira a seguir!



<https://aposti.la/p3fcc>

1. Primeiro, abra o link do QR code acima no seu navegador, então, você precisa adicionar no mínimo 3 novos cenários (caso você não se lembre como fazer isso, você pode adicionar pelo botão “Selecionar cenário” no lado direito inferior da página).
2. A seguir, vá para a categoria “Controle” e adicione o bloco “se então” no final dos 2 conjuntos, um no código para ir para esquerda e outro para ir para a direita.
3. Agora, vá para a categoria “Sensores” e adicione novamente nos 2 conjuntos o bloco “tocando em 'ponteiro do mouse'” e, após isso, altere o valor “ponteiro do mouse” para “borda”.
4. Agora, precisamos adicionar o bloco “próximo cenário” que está na categoria “Aparência”, coloque ele dentro do bloco “se então” dos 2 conjuntos.
5. Agora vá para a categoria “Movimento” e procure pelo bloco “vá para x: y:”, coloque ele abaixo do bloco “próximo cenário” e digite nos 2 espaços, primeiro o valor “-185” e no segundo o valor “-100”.

Saudações! Aqui é o sapo mago! E desta vez eu serei o seu arauto do conhecimento, pois se tem alguém que é realmente apaixonado por livros nessas terras, este alguém sou eu.

Neste capítulo da apostila, espero poder trazer luz a seu conhecimento sobre programação, pois aqui falaremos mais sobre os operadores relacionais.



9.1. Operadores Relacionais

O Sol é maior que a Lua? A Lua brilha mais que o Sol? A Terra tem o mesmo tamanho que Marte? Tudo isso são perguntas que testam a relação entre as coisas, por meio de comparações simples!

Na programação isso não é diferente! Sempre fazemos comparações em códigos, principalmente quando queremos testar coisas. E assim como eu e os livros, as Estruturas de Controle, sempre andam de mãos dadas com os Sensores, os Operadores Relacionais, ou ainda, os Operadores Lógicos, que veremos futuramente.

MAS... Falando sobre os operadores relacionais, eles se organizam em 3. E assim como os nossos operadores matemáticos, eles comparam números e seus valores ou caracteres! E como exatamente nós sabemos os resultados dessas comparações? Bem, os operadores relacionais sempre e em todos os resultados, nos retornam um dos dois seguintes valores: "true" ou "false". Com o resultado retornado por ele sendo "true", que é "verdade" em inglês,

significando que sim, era verdade a afirmação que ele testou.

Ou ainda com o resultado declarado por ele sendo "false", que significa em inglês "falso", indicando que a declaração em questão também era falsa.

Bem simples, não?

O primeiro Operador Relacional que temos no Scratch, é o operador "maior que", representado pelo seguinte símbolo: $>$. Com seu bloco no Scratch sendo o seguinte:



Este operador testa se o primeiro número digitado, no caso o da esquerda, é maior que o segundo, no caso o da direita. E como vimos antes, ele retornará "true" se isso for verdade, ou "false", se isso for falso.

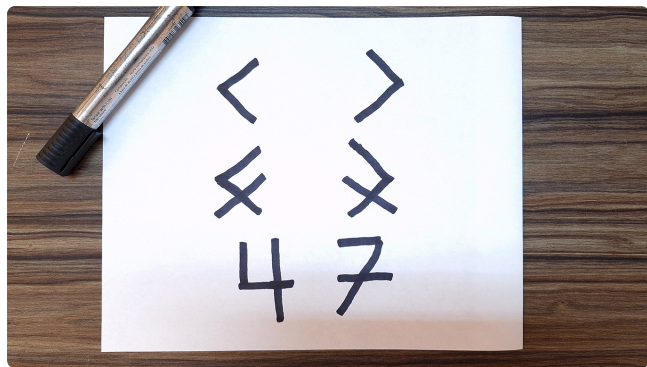
Já o segundo Operador Relacional é o operador "menor que" representado pelo seguinte símbolo: $<$. Entretanto, veja a seguir, a sua representação no Scratch é quase igual a que vimos acima, apenas mudando o símbolo do centro do bloco.



E ele é quase idêntico a sua versão "maior que", com a única diferença sendo que ele é o bloco "menor que"! Então ele testa se o número que está na caixa da esquerda é menor que o número que está na caixa da direita.

E sim, estou ciente de que eles são muito semelhantes. Por isso veja, se desenharmos em uma folha de papel o seus símbolos, " $<$ $>$ " e colocarmos um risco em cada uma das perninhas

de baixo deles, nós teremos os respectivos números 4 e 7. Veja no exemplo a seguir.



E assim como o número 4, é menor que o número 7, "<" é o símbolo de "menor que" e ">" é o símbolo de "maior que".

E, o grand finale vai para o operador "igual a", que usa o tradicional símbolo "=". Este operador apenas vai retornar verdadeiro se o número ou palavra que estiver na caixa da esquerda for exatamente igual ao número, ou palavra que estiver no bloco da direita.

Veja como ele é no Scratch. Ah, você deve se lembrar dele das aulas anteriores.



Abra! Cadabra! Pé de cabra! Vamos fazer nosso exercício?

9.2. Exercícios

No exercício desse capítulo, nós continuaremos desenvolvendo o código que fizemos na nossa aula. Contudo, entretanto, todavia, faremos alguns incrementos!

Então se certifique de baixar ele através do seguinte link:



<https://aposti.la/p3e20>

Bem, vamos começar. Nosso objetivo é bem simples, nós vamos criar uma atualização que calcule a média de 3 notas ao invés de uma nota, e informe ao nosso usuário, se o aluno passou ou não no semestre.

1. E para isso vamos ter de mudar o nome da nossa variável "nota" para "nota final". Nós podemos fazer isso clicando em Variáveis, e após clicar com o botão direito na nossa variável, escolher a opção "Renomear variável".

2. Ainda nessa categoria criaremos mais 3 variáveis, chamaremos elas de "nota 1", "nota 2", e "nota 3".

3. E então nós adicionaremos, indo à categoria "Sensores", o bloco "pergunte 'Qual o seu nome?' e espere" 3 vezes, logo abaixo do bloco "mude 'nome' para 'resposta'".

4. Você também deve mudar o conteúdo desses 3 blocos para "Digite a nota 1:", "Digite a nota 2:" e "Digite a nota 3:", nessa ordem mesmo.

5. E logo após isso, voltando a categoria Variáveis, você deve colocar, também 3 vezes abaixo de cada um dos blocos de "pergunte", o bloco "mude 'my variable' para '0'".

6. E então, você muda o "my variable" para as 3 variáveis novas que você criou, "nota 1", "nota 2" e "nota 3".

7. Agora, vamos mudar o "0" desse bloco, para isso vamos à categoria Sensores.



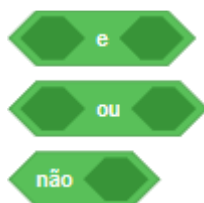
Olá, e seja bem-vindo(a) ao décimo capítulo da nossa apostila, você já passou da metade das aulas e apostilas, meus sinceros parabéns por chegar tão longe!

Em nossa apostila de hoje, veremos mais a fundo uma explicação sobre os Operadores, para o caso de você não ter entendido muito bem durante a aula.



10.1. Operadores Lógicos

Existem três Operadores Lógicos no mundo da programação, sendo eles o "AND" o "OR" e o "NOT", que respectivamente suas traduções são: O "E" que vimos na aula dessa apostila; O "OU", e também o "NÃO", que veremos na próxima aula!



Falando mais sobre o Operador "E", ele é um operador de comparação, como eu falei durante o curso, e a função dele no código é fazer comparações lógicas, e também, como foi falado durante o curso, você pode adicionar quantos blocos "E" quiser, e, fazendo isso, você estará aumentando o número de condições que deverá

ser verificada antes da execução do código principal, algum código "se" então!

Sendo assim, você terá um código que verificará "se 50 = 50 e "minha variável" = 2 e "morango" = sim", por exemplo.

E após fazer a sua comparação, ele te devolverá um valor Booleano, isso que vimos lá no começo do nosso curso, e já que citei isso novamente, vamos lembrar o que é um valor Booleano.

Booleano é um valor representado por "false" ou "true", (falso ou verdadeiro), isso de uma forma bem resumida, sendo assim, como acabei de falar, o bloco/código fará as comparações, e então retornará um valor Booleano!

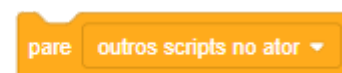
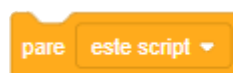
10.2. Bloco "pare"

Hoje vimos esse bloco pela primeira vez, então, nada mais justo que uma explicação mais profunda sobre ele, não?



Então, ele é um bloco que está presente na categoria "Controle", então, ele tem o papel de controlar o código e, como o próprio nome sugere, o que ele faz no código é parar, então, a partir dele, podemos parar a execução de qualquer código que quisermos.

Além de conseguir parar a execução de "TODOS" os códigos, de todos os atores, também podemos parar de atores específicos, ou até mesmo códigos específicos!

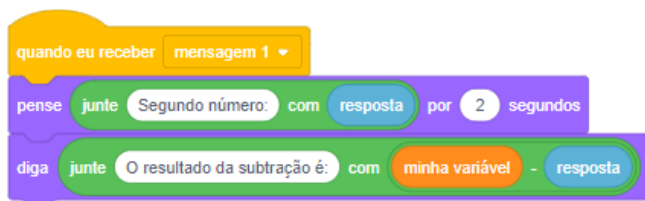




Veja o código acima, esse código é referente à aula 6, foi o projeto feito na nossa prática, era o código de uma conta de subtração, caso você não se lembre.

Agora, observe que, antes do bloco "pare", eu adicionei um bloco "transmita", pois logo após esse bloco está o bloco "pare".

E aqui está a continuação do bloco acima, se você decidir colocar aquele bloco "pare" usado acima, você pode utilizar, por exemplo, o bloco "transmita" para continuar executando o código em outro conjunto, ou até mesmo em outro ator.



E por falar nisso, o bloco pare está indicando "este script" para ser parado, isso indica que o código vai terminar ali e não pode ser adicionado nada abaixo do bloco "pare", como você mesmo pode ver, pois ele não tem uma "perninha" para encaixar outro bloco, ele tem a base plana.

Por outro lado, o bloco "pare 'outros scripts no ator'" fará com que os outros códigos que estejam sendo executados juntamente do principal, também parem!

10.3. Exercícios

Em nosso exercício de hoje, você irá criar de maneira simples o código que viu durante a aula 7, que era o código para fazer o Robô atravessar a rua, porém utilizando o Scratch como ator! Se estiver com dificuldade, não se esqueça de olhar o gabarito!

1. Crie um novo arquivo, adicione o cenário "Urban" ou "Mural" e também adicione o ator "Convertible" e "Convertible 2", ajuste o Scratch para a parte de baixo do projeto, e cada um dos carros em cada lado da rua, um do lado esquerdo e outro no direito.

2. Com o Scratch selecionado, vá para a categoria "Eventos" e adicione o bloco "quando 'ir' for clicado", vá para a categoria "Movimento", arraste o ator Scratch para onde você queira que ele comece o projeto, e então arraste o bloco "vá para x y" e confira se os números do bloco estão de acordo onde o ator está no momento!

3. Agora, vá para a categoria "Sensores" e adicione um bloco de pergunta abaixo do bloco de movimento e escreva nele "Posso atravessar a rua?". Vá para a categoria "Controle" e adicione o bloco "se então senão" abaixo do bloco de pergunta. Vá para a categoria "Operadores" e adicione dois blocos "E" e coloque um dentro do outro, depois adicione eles para dentro do bloco "se então senão", e agora, adicione o bloco "=" em cada espaço vazio nos blocos "E".

4. Retorne até a categoria "Sensores" e adicione o bloco "resposta" dentro do primeiro bloco "=". e escreva "sim", no espaço em branco desse bloco. Vá para a categoria "Movimento" e adicione o bloco "deslize por 1 segs. até x y" no primeiro espaço do bloco "se então senão", e então, defina o valor de X e Y para onde o ator deva parar quando for atravessar a rua, para isso você pode arrastar o ator ou algum carro até o local que você queira, e então pegue os valores X e Y atuais dele, depois bote no lugar que estava.

5. Agora, vá para a categoria "Aparência" e adicione dois blocos "diga 'Olá!' por 2 segundos",



Olá, pessoal! É a Giga aqui na área! E nesse capítulo nós continuamos a falar sobre Operadores Lógicos, que assim como você pode ver na aula, não são nada complicados!

Na realidade, é importante dizer isso, programação não é complicada! Eu pelo menos acredito nisso: para mim, programação é igualzinha a sorvete! Sim, isso mesmo, sorvete!

Mas, por quê? Bem, sorvetes são deliciosos! Mas se a gente tomar eles muito rapidamente dá uma dor de cabeça danada! E programação é exatamente assim, nós temos que "tomar" ela devagar! Aproveitando todo o conhecimento e as lições na medida certa, para evitar fortes dores de cabeça! Hehehe!



E os usos disso? Bem, são infinitos! Nós podemos imaginar um seguinte cenário que você já deve ter visto!

Imagine que você quer atravessar a rua! E sim, você vai ver esse exemplo com outros olhos agora hehehe... Para atravessar a rua, nós temos que ter certeza que não tem carro nenhum passando na rua, certo? E como você viu, para isso nós fazemos testes! Agora imagine que essa rua é uma rua de mão dupla, ou seja, vem carro dos dois lados!

Se vir um carro da direita, OU da esquerda, você não atravessa! Outra forma de ver isso, é imaginar que o lado da direita E o lado da esquerda não podem ter carro nenhum passando!

Agora eu vou dizer algo que é bem a cara do Sapo Mago, na programação não existe apenas uma solução para um problema, existem muitas soluções para um mesmo problema! Só que nem sempre essas outras soluções vão ser fáceis de se achar, e às vezes nem são as melhores mesmo.

E o operador OU vem para isso, para abrir uma janela de possibilidades maior e facilitar nossa vida resolvendo esses problemas.

11.1. Operador Lógico "OU"

Assim como vimos na aula passada, o operador "OU" se parece e muito com o operador "E" em estrutura, mas, com a grandíssima diferença de retornar "true", desde que pelo menos um dos seus dois testes lógicos seja verdadeiro.



Eis aqui um belo exemplar de um operador "OU".



11.2. Exercícios

No exercício desse capítulo, nós iremos me fazer, isso mesmo! Euzinha, comer alguns bolos no Scratch!

1. Fazer isso é bem, beem fácil! Vamos começar abrindo o Scratch no seu computador, como sempre fazemos.

2. Depois disso, vamos clicar em "Selecione um Ator", e na tela que abrir procure por Giga, e clique em Giga Walk.

3. Nós vamos utilizar essas fantasias para fazermos o nosso código, então podemos apagar o conjunto onde está o Scratch, clicando na lixeirinha acima da cabeça dele.

4. Após isso, estando com o ator Giga Walk selecionado, clicamos em fantasias, e, com as fantasias selecionadas, vamos duplicar cada umas das três opções.

5. As novas opções devem estar nomeadas como "Giga walk4, Giga walk5 e Giga walk6", você deve então clicar no botão de espelhar horizontalmente em cada uma delas.

6. Depois disso, vamos a guia "Código", e lá vamos até a categoria de "Eventos".

7. E então, clicamos e arrastamos o bloco "Quando a tecla 'espaço' for pressionada", até a área de edição do código.

8. Nós devemos clicar então na caixa "espaço" de dentro desse bloco e mudar para "seta para direita".

9. Após isso, vamos até a categoria "Movimento" e arrastamos o bloco "mova '10' passos" até abaixo do último bloco que posicionamos.

10. Depois voltamos para a categoria "Controle" e arrastamos o bloco "espere '1' seg" até abaixo do último bloco que a gente posicionou!

11. Depois disso vamos para a categoria "Aparência" e arrastamos até abaixo desse bloco

o bloco "mude para fantasia 'Giga walk6'", ah, pode ser que o nome da fantasia seja diferente, então cuidado ao escolher o bloco!

12. Agora nós clicamos com o botão direito em cima do bloco "mova '10' passos", e selecionamos a opção "Duplicar".

13. O conjunto de 3 blocos que aparecer nós colocamos abaixo do que já existe.

14. Repetimos o processo mais uma vez! Devemos poder enxergar 3 blocos "mude para fantasia 'Giga walk6'", e vamos mudar o "Giga walk6" de cada um para "Giga walk1", Giga walk2" e "Giga walk3".

15. Depois disso, alteramos os valores de cada bloco "espere '1' seg" para "espere '0.1' seg". É importante que você teste seu código agora, se você apertar a seta direita do seu mouse, a nossa belíssima atriz deve se mover!

16. E então após isso nós devemos adicionar mais alguns blocos, vamos começar indo para a categoria de "Controle" e lá nós iremos arrastar o bloco "se então" até embaixo do conjunto de blocos que criamos.

17. Vamos então até a categoria de "Operadores", e nela vamos arrastar até dentro do favinho de mel do bloco "se então" o bloco operador "OU".

18. E dentro de cada um dos dois favinhos de mel que tem dentro do bloco "OU", nós vamos colocar um bloco "tocando em 'ponteiro do mouse?'," que a gente acha na categoria de "Sensores".

19. E por fim, para dentro do bloco "se então" na parte interna que está vazia, nós vamos colocar um bloco "diga 'Olá!' por '2' segundos" que a gente acha na categoria "Aparência". Nele nós substituiremos o "Olá!" por "Nhack!".

20. Calma que já estamos acabando!

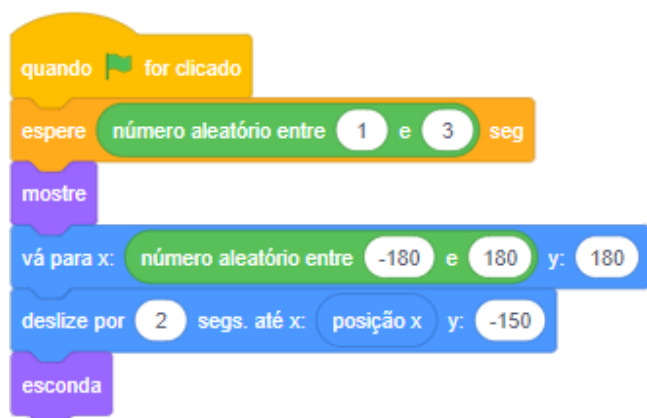
21. Vamos duplicar esse código todo agora, clicando com o botão direito no topo dele, que no caso é o bloco "quando a tecla 'seta para direita"

Olá, seja bem-vindo(a) à mais um capítulo da nossa apostila, e hoje, vou lhe explicar para que serviram os blocos que usamos durante nossa aula, no caso, a estrutura que usamos!



Bom, como você já deve saber, o objetivo da aula era criar um contador de pontos, dito isto, criamos um bem simples utilizando o que já tínhamos aprendido em aulas anteriores.

Então, vamos começar com o primeiro conjunto de blocos que criamos! Observe ele a seguir.



Esse código era responsável por fazer nosso peixe aparecer na nossa tela e fazer ele deslizar até o fundo do mar!

12.0.1. Blocos "quando ir for clicado" e "espere"

Começamos com o bloco “quando 'ir' for clicado”, que é a bandeirinha verde. Quando clicamos na bandeira, há um bloco “espere”, e dentro dele colocamos um bloco de “número aleatório”. Isso faz com que o peixe espere entre 1 a 3 segundos antes de aparecer na tela.

12.0.2. Blocos "mostre", "número aleatório" e "vá para x: y:"

Em seguida, temos o bloco “mostre”. Ele faz com que o peixe apareça na tela se ele estiver invisível. Depois de mostrar o peixe, usamos o bloco “vá para x: y:”. Esse bloco define onde o peixe vai começar. Para escolher um lugar aleatório para o peixe, colocamos o bloco “número aleatório” no valor X do bloco “vá para”. Isso significa que o peixe vai aparecer em um lugar aleatório na linha horizontal (eixo X). O valor Y, que é a linha vertical (de baixo para cima), é fixo e positivo. Então, o peixe vai aparecer no topo da nossa tela de jogo.

12.0.3. Blocos “deslize”, “posição x” e “esconda”

Também usamos o bloco “deslize”. Esse bloco faz o peixe deslizar do ponto inicial (o valor X) até o ponto final (o valor Y). O número de segundos que colocamos no bloco é o tempo que o peixe levará para chegar ao ponto final. O ponto inicial X é a “posição x”, que foi definida pelo bloco “número aleatório” que usamos antes. A posição Y é um valor fixo negativo, o que faz o peixe deslizar até o fundo do mar.

Por último, vamos falar sobre o bloco “esconda”. Esse bloco faz o peixe desaparecer da tela. É um bloco muito útil em muitos projetos diferentes.



Agora, vamos explicar o segundo conjunto de blocos. Esse conjunto tem outra função e trabalha junto com o primeiro conjunto. A função desse conjunto é fazer o peixe mágico receber um sinal se o Scratch, nosso ator, tocar no peixe. Porém, esse código só verifica isso quando o ator está se movendo. Talvez você tenha notado isso durante os testes na aula.

Se você percebeu isso, saiba que era um “erro” proposital, porque com o que aprendemos até agora, essa era a melhor maneira de criar o código. Mas não se preocupe, pois mais adiante vamos atualizá-lo!

Então, vamos lá.

12.0.4. Blocos “quando eu receber” e “se ” então”

Então, iniciando o código, temos o bloco “quando eu receber 'mensagem 1'”, esse bloco é semelhante ao bloco da bandeirinha verde, porém, ele só ativará caso o bloco “transmita 'mensagem 1'” seja ativado.

Então, ao ser ativado, ele faz o restante do código ser lido. A seguir, nós temos dois blocos “se ” então”, que caso você não se lembre, esse bloco é responsável por iniciar uma verificação, então, caso ela for verdadeira, ele executa o que está dentro dele, e caso for falsa, ele não faz nada!

12.0.5. Blocos “ou” e “menor que”

Como sabemos, o bloco “se ” então” realiza verificações. Dessa vez, usamos os blocos “ou” e “menor que”. Primeiro, o bloco “ou” verifica duas condições e vê se uma delas é verdadeira. É por

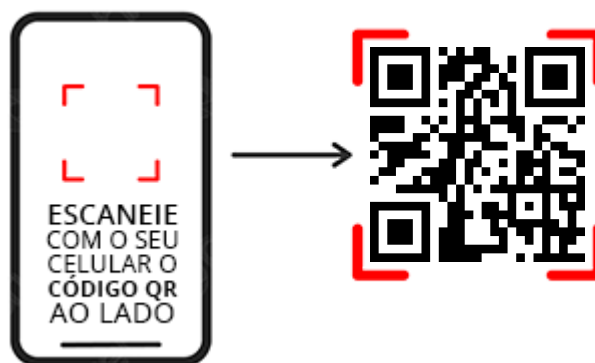
isso que ele se chama “ou”, pois ele verifica "ou uma coisa, ou outra". No nosso código, o bloco “ou” verifica se o peixe está tocando no ator ou se está no fundo do mar.

Depois, temos o bloco “menor que”. Esse bloco verifica se o primeiro número é menor que o segundo e nos diz se isso é “verdadeiro” ou “falso”. No nosso código, colocamos o bloco “menor que” dentro do bloco “ou”. Então, a verificação funciona assim: ele verifica "se o peixe está tocando no 'ator 1' ou se o peixe está no fundo do mar", que no caso é representado pelo código "posição y < -149".

Para finalizar, temos outro bloco "se " então" verificando novamente se o ator está tocando no peixe, que caso essa verificação seja 'verdadeira', ganhamos pontos, e se caso seja 'falsa', o peixe irá simplesmente desaparecer.

12.1. Exercícios

O exercício dessa apostila será criar um código utilizando blocos que já utilizamos durante as aulas, mais especificamente, você deverá acessar o link abaixo ou ler o qr code, então, você terá acesso a um projeto que contém um código de pulo, então, a partir dele, você deverá seguir o passo a passo do exercício.



<https://aposti.la/5o757>

1. Após acessar o qr code ou link, vamos começar com a movimentação do ator, para isso, vá até a categoria "Eventos", adicione 2 blocos "quando a tecla 'espaço' for clicada" até a área de

código, e em seguida, mude o valor desses dois blocos, colocando um para verificar um clique para a seta esquerda do teclado e o outro para a seta da direita.

2. Agora, vá até a guia "Fantasias" do ator, duplique as duas fantasias que já estão prontas, inverta as duas novas e mude para um nome de sua preferência, pois elas serão as fantasias que farão o nosso ator virar para a esquerda.

3. Retorne até a guia de "Código", e então, faça a seguinte sequência nos blocos novos que você adicionou: vá para a categoria "Aparência" e procure pelo bloco "mude para a fantasia", após achá-lo, coloque esse bloco duas vezes no código para o ator andar para a esquerda, repita o mesmo processo para a direita.

4. Troque os valores dos códigos "mude para a fantasia", de acordo com os nomes que você deu e seus respectivos lados, e então, vá para a categoria "Controle" e adicione dois blocos "espere 1 seg" entre os dois blocos "mude para a fantasia", em cada um deles, coloque o valor "0.05", repita o processo para o lado direito também.

5. Agora, vá para a categoria "Movimento" e procure pelo bloco "adicione '10' a x", então, adicione ele entre os dois blocos "espere", e repita o processo para o lado direito.

6. Comandos concluídos, agora você precisa ir até a categoria "Variáveis", chegando lá, crie uma nova variável com o nome "Pontos", e então, arraste o bloco "adicione 1 a variável" no fim do conjunto que faz o ator pular. Mude a variável que irá receber o valor do bloco "adicione". Após isso, vá para a categoria "Eventos".

7. Agora, arraste o bloco "transmita 'mensagem 1'" para baixo do bloco "adicione", e após isso, arraste o bloco "quando 'ir' for clicado" para o lugar que você quiser, e em seguida, arraste o bloco "mude 'Pontos' para 0" para baixo do bloco "quando 'ir' for clicado", esse bloco se encontra na categoria "Variáveis".

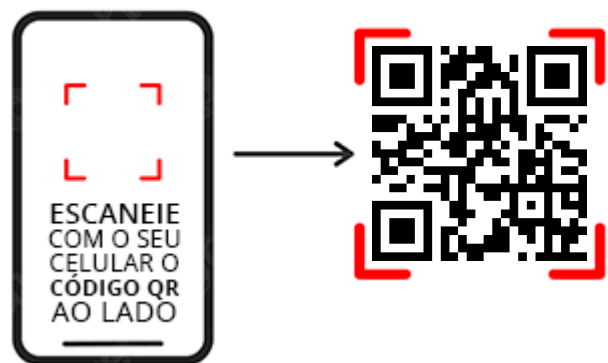
8. Agora vá para a categoria "Eventos" novamente, e arraste o bloco "quando eu receber

mensagem 1" para qualquer lugar que você queira, pois agora vamos criar um código para fazer o gato miar.

9. Vá para a categoria "Controle" e arraste o bloco "se " então" para dentro do novo conjunto que você acabou de criar. Após, vá para a categoria "Operadores", e então, arraste primeiramente o bloco "=", para dentro do "se " então", e após ter feito isso, arraste o bloco "número aleatório entre 1 e 10" para dentro do primeiro espaço do bloco "=", coloque no lugar do número '10' algum valor menor, para ter mais chances do nosso ator miar, e também escolha um número para colocar no segundo espaço do bloco "=", porém, esse número deve estar entre o número 1 e o número que você digitou no segundo espaço do bloco "número aleatório".

10. E por fim, vá para a categoria "Som" e arraste o bloco "toque o som 'Miau'" para dentro do bloco "se " então"!

Pronto, exercício finalizado, caso você não tenha conseguido fazer, verifique os arquivos auxiliares, gabaritos ou acesse o link abaixo!



<https://aposti.la/zzb1s>



Bip Bip Bip Bop! Seja bem-vindo(a) ao capítulo 13 do seu curso de Programação Kids, e neste capítulo "do balacobaco" nós vamos aprender mais e mais sobre Estruturas de Repetição!



13.1. Estruturas de Repetição

O que são? Essa pergunta eu te respondo!

Elas são estruturas que repetem os códigos que estão dentro delas! E nós definimos quantas vezes isso acontece.

Imagine comigo agora, você deseja programar um robô para fazer uma caminhada, você sabe o destino do robô, digamos que seja uma vendinha na esquina da sua casa!

Você, sendo um programador ou uma programadora muito dedicada, contou quantos passos você deu até chegar na vendinha, e programou o robô para dar essa mesma quantidade de passos. Ele vai andar e andar, até chegar na vendinha, pois você disse para ele exatamente quantos passos ele teria que dar até chegar lá.

Agora imagine o seguinte, digamos que você mude de vendinha, e que essa vendinha seja

longe para chuchu, simplesmente não teria como você contar os passos até ela, pois você se perderia na contagem. É nesse cenário que usamos as estruturas de repetição, pois com elas você simplesmente poderia dizer "Ei robô, dê passos até chegar na vendinha", e ele chegaria, não importa o quão longe fosse a vendinha.

E é assim que elas funcionam, a gente estabelece um limite, ou não, e elas seguem ele!

Vamos ver agora as principais Estruturas de Repetição!

13.2. Repita 10 vezes

Este bloco repete uma quantidade definida de vezes tudo o que está dentro dele, você pode alterar o valor digitando, ou posicionando uma variável numérica dentro do bloco!



13.3. Sempre

Este bloco repete para sempre todos os códigos dentro dele, mas, apenas quando ele é ativado! Contudo, nós podemos usar um outro bloco que para esse tipo de comando, como o bloco "pare 'todos'" que já vimos anteriormente, ou, simplesmente, apertar o botão "Pare", que fica ao lado do botão "Ir".



13.4. Repita até que

Peti e Repeti, esse bloco vai repetir até que uma coisa seja verdadeira, como, por exemplo, você pode usar algum bloco sensor, como o "tocando em 'ponteiro do mouse'?", que o bloco "repita até que" irá repetir todos os comandos que estiverem dentro dele, até que este teste seja "true".



13.5. Exercícios

Vamos fazer o nosso ator, que serei eu, a fazer *DRIFT!*



Bem, de maneira segura e controlada, obviamente! E seguindo o mouse!

1. Começaremos abrindo o nosso Scratch de cada dia, em um novo documento! O que já deve ser bem fácil para a gente.

2. Em seguida, vamos até "Eventos"! De lá nós arrastaremos dois blocos "Quando 'lr' for clicado", os posicione como achar melhor na tela!

3. Na sequência vamos até "Controle"! De lá pegaremos dois blocos "sempre", e os posicionaremos um abaixo de cada bloco "quando 'lr' for clicado".

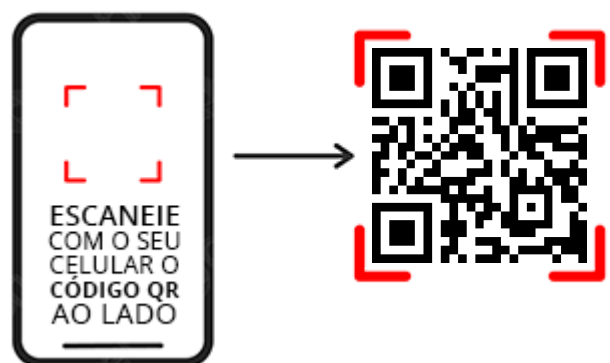
4. Agora vamos clicar na categoria "Movimento". De lá nós vamos clicar e arrastar um bloco "aponte para 'ponteiro do mouse'" para dentro do "sempre" do primeiro conjunto de "quando 'lr' for clicado" e "sempre".

5. E, depois disso, arrastaremos um "deslize por '1' segs. até 'posição aleatória'" até o "sempre" de dentro do segundo conjunto.

6. E por fim, isso mesmo, já está acabando! Mudaremos os valores de "deslize por '1' segs. até 'posição aleatória'", clicando no "1" e mudando ele para "0.5" e "posição aleatória" para "ponteiro do mouse".

7. Ah, é isso mesmo o nosso código, você ainda pode regular a "velocidade" do ator, simplesmente mudando o "0.5" do bloco "deslize por '0.5' segs. até 'posição aleatória'", sendo que valores mais baixos aumentam a velocidade e maiores diminuem.

Faça seus experimentos! Bem, deixaremos aqui um QR Code e um link para você jogar esse jogo ou ainda verificar se errou alguma coisa no código. Use ele à vontade!

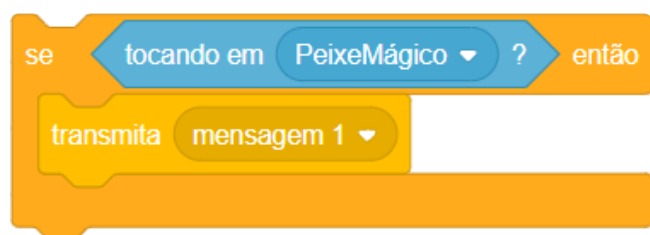


<https://aposti.la/4dqi3>

Olá, seja bem-vindo(a) a mais um capítulo de apostila, e hoje contamos com o conteúdo da aula 14, ou seja, na apostila de hoje, além da apresentação de um bloco novo, você também verá uma explicação do que vimos e fizemos durante a aula.



Começando pelo conteúdo da aula, vamos ver por partes. Começando pelos dois conjuntos de códigos que apagamos antes de automatizar os códigos do peixe mágico.

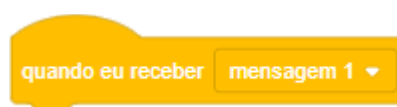


Conjunto que apagamos.

Bom, antigamente eles serviam para enviar um sinal até outro conjunto, que por sinal, está presente no peixe mágico. O sinal era enviado basicamente em todo momento que nosso ator se movia, e por conta de sua forma primitiva, nós o excluímos para deixar nosso código mais limpo, pois ele não teria mais utilidade!

E avançando para os conjuntos presentes no peixe mágico, vamos começar pelo conjunto que acabamos de mencionar, o conjunto que recebia o sinal quando nosso ator se movia e encostava no peixe ao mesmo tempo, e que também era responsável por adicionar nossa pontuação!

Ele teve uma pequena evolução, perdendo o bloco:



Bloco perdido.

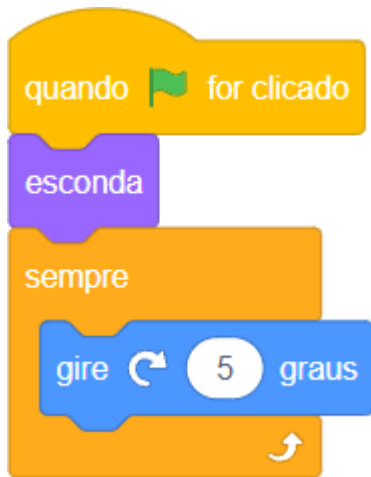
Mas em compensação acabou ganhando esses 3 blocos em seu lugar:



Bom, tirando o bloco de clique na bandeira e o de mudar os pontos para zero, o bloco "sempre" foi o nosso salvador nesta aula, sendo responsável principalmente por "automatizar" nosso código, pois como você já deve ter percebido durante a aula, ele "sempre" executará o código que está dentro dele, ao menos que pausemos a execução do código ou através do bloco "pare"!

Além dessa automação, também criamos um conjunto que era responsável por fazer o

peixe ficar rodando para sempre!



E por último, automatizamos o conjunto que fazia o peixe ficar aparecendo novamente sempre depois que desaparecia após ser pego ou por não conseguirmos pegá-lo, que foi bem simples, apenas adicionamos o bloco “sempre” no conjunto que já tínhamos pronto.

14.1. Bloco "espere até que"

O bloco “espere até que” é um bloco de controle. Isso significa que ele ajuda a controlar quando as coisas acontecem no seu projeto. Esse bloco faz o seguinte:

- Ele espera até que uma condição seja verdadeira.
- Depois que a condição é verdadeira, ele continua com o resto da programação.

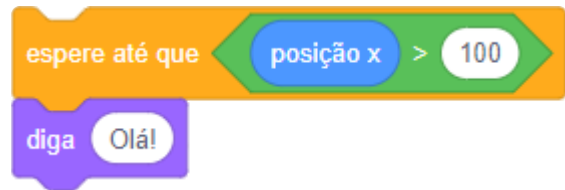
14.1.1. Como usar o bloco "espere até que"

Imagine que você quer que um personagem do seu jogo ande até uma certa posição e depois diga “Oi”. Você pode usar o bloco “espere até que” para garantir que ele só diga “Oi” depois de chegar nessa posição.

Aqui está um exemplo de como usar:

1. Coloque o bloco “espere até que” no seu script.
2. Dentro do bloco, coloque uma condição. Por exemplo: “posição x > 100”.
3. Adicione o que deve acontecer depois que a condição for verdadeira. Por exemplo: “diga 'Olá!'”.

Então o código ficaria assim:



Isso significa que o personagem vai esperar até que sua posição x seja maior que 100. Só depois disso ele vai dizer “Oi”.

O bloco “espere até que” é muito útil quando queremos que algo aconteça apenas depois que uma certa coisa acontecer. Ele ajuda a controlar a ordem das ações no seu projeto, basicamente.

14.2. Exercício

Para o exercício dessa apostila, você fará um joguinho simples estilo “Caça a Toupeira”, onde você precisa clicar no personagem para ganhar pontos. E a toupeira será a Giga, hehe, então trate de clicar em todas as Gigas que você achar!

14.2.0.1. Conjunto 1

Esse conjunto servirá para tornar aleatória a aparição da Giga, ou seja, onde ela vai aparecer e até quando ela ficará repetindo o código.

1. Começando a partir de um arquivo em branco, selecione a Giga como ator e apague o Scratch.
2. Agora vá para a categoria “Eventos” e arraste o bloco “quando 'ir' for clicado”, a famosa bandeirinha, para a área de trabalho.
3. Vá para a categoria “Controle”, e arraste o bloco “repita até que” para baixo do bloco “quando 'ir' for clicado”.
4. Agora vá para a categoria “Operadores” e adicione dentro do bloco “repita” o bloco “menor que” (<).
5. Agora, vá para a categoria “Variáveis” e crie duas variáveis novas, uma com o nome “Pontos” e outra com o nome “Tempo”.
6. A seguir, pegue a variável “Tempo” e coloque dentro do espaço em branco no bloco “menor que”, e já altere o valor de '50' para '1'.

7. Então, vá para a categoria “Aparência”, procure e arraste o bloco “mostre” e então arraste ele para dentro do bloco “repita até que”.

8. Agora vá para a categoria “Controle” e adicione o bloco “espere '1' seg” abaixo do bloco “mostre”.

9. A seguir, vá para a categoria “Movimento” e adicione o bloco “vá para 'posição aleatória” em baixo do bloco “espere '1' seg”.

14.2.0.2. Conjunto 2

Esse conjunto será responsável por definir o tempo que o código será executado, ou seja, será um cronômetro. Além de também ser responsável por zerar a pontuação depois de cada jogatina.

1. Vá para a categoria “Eventos” e adicione o bloco “quando 'ir' for clicado” em um lugar afastado do último conjunto.
2. Agora vá para a categoria “Variáveis”, então, adicione o bloco “esconda a variável”, nesse bloco deve estar a variável inicial, que é criada junto com o projeto, pode apagá-la se quiser, pois não vamos utilizar ela, e em seguida selecione a variável “Pontos” no bloco “esconda a variável”.
3. A seguir, arraste duas vezes o bloco “mude” da categoria “Variáveis” para baixo do bloco “esconda”, no primeiro bloco coloque a variável “Pontos” e deixe o valor como '0', e no segundo bloco coloque a variável “Tempo”, e o valor você mesmo pode colocar, pois esse será o valor que o jogo vai durar.
4. Agora será bem simples, volte até o primeiro conjunto que você criou e duplique o bloco “repita até que”, ele deverá vir com os 3 blocos que adicionamos dentro dele anteriormente, porém, apenas vamos precisar do bloco “espere '1' seg”, então, duplique ele, a seguir, exclua os códigos que estão dentro dele, deixando apenas o bloco “espere” e então, coloque ele no final do conjunto que estamos fazendo agora.
5. Então, já que temos a estrutura quase finalizada, adicione o bloco “adicione '1' a 'variável” para baixo do bloco “espere 1 seg”, esse bloco adicione será responsável por diminuir o tempo do nosso cronômetro, juntamente com o bloco “espere”, porém, para ele funcionar corretamente, antes do número '1', coloque o símbolo de “.”.

6. E finalizando esse conjunto, adicione o bloco “mostre a variável” no final de todo o conjunto, e após isso, mude a variável selecionada neste bloco para a variável “Pontos”.

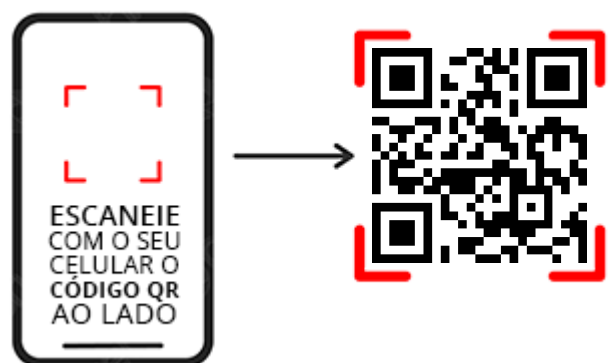
14.2.0.3. Conjunto 3

E para finalizar, esse conjunto vai servir para adicionar um evento de clique, isso, quando clicarmos na Giga, será um código bem simples, mas funcional!

1. Para começar, em outro lugar afastado dos outros 2 conjuntos que você criou até agora, vá até a categoria “Eventos”, procure e adicione o bloco “quando este ator for clicado” até a área de código. Este bloco é autoexplicativo, ele será executado sempre que clicarmos no ator, mesmo que o código não tenha sido iniciado através da bandeira.
2. Agora, vá até a categoria “Variáveis” novamente, e adicione o bloco “adicione '1' a variável”, e então mude a variável que está selecionada para a variável “Pontos”.
3. Por último, vá até a categoria “Aparência” e procure pelo bloco “esconda”, então adicione ele abaixo do bloco “adicione '1' a 'Pontos”.

Se você chegou até aqui, parabéns!

O código está finalizado, porém, está na dificuldade 'fácil' ainda, se quiser deixar ele mais frenético e difícil, duplique a Giga e, no ator que foi e será duplicado, caso você tenha duplicado a Giga mais do que uma vez, exclua todo o “Conjunto 2” que você criou.



<https://aposti.la/nnv7h>



Olá, seja bem-vindo(a) a mais uma aula do seu curso de Programação Kids! Neste capítulo nós vamos continuar aprimorando nosso código e também vamos rever alguns blocos muito interessantes, e adicionar uma função nova no nosso projeto!

15.1. A variável "fantasia 'nome'/'número'"

O grande divisor de águas nessa aula, além é claro do bloco "sempre" que permite uma verificação automatizada de qualquer "coisa", é a própria "coisa" que está sendo constantemente verificada.

A variável "fantasia 'número'" e a sua variável irmã, "cenário 'número'", possuem uma função muito importante, entregar para a gente uma forma de diferenciar os cenários e as fantasias dos nossos atores. Existem diversas combinações possíveis onde podemos usar esses blocos, e em diversos testes, como, por exemplo, caso estejamos querendo testar exatamente um nome de uma fantasia específica, basta arrastar essa variável para dentro do primeiro espaço do bloco "igual a", representado pelo símbolo de "=" e digitar o nome ou o número que você deseja testar.

Eis logo abaixo um belo exemplar de cada bloco destes, encontrados em seu habitat natural, o próprio Scratch.

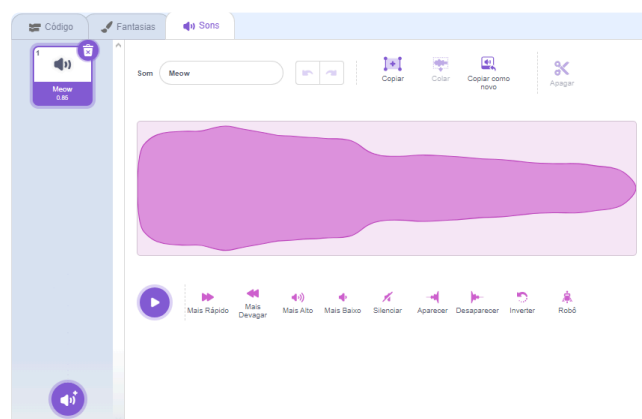


Mas, e caso a gente apenas queira testar um elemento específico? Nem sempre haverá funções extremamente específicas como uma variável para fantasias, no Scratch nós até temos, mas em outras linguagens de programação, nós temos que criar essas variáveis e configurar os seus valores, tal como muitas outras coisas e é aí que entra a melhor ferramenta de todas:

A nossa cachola! Programar é sobre usar a cabeça, ser criativo e acima de tudo, errar. Pois erros ensinam tanto quanto acertos e além disso, eles ensinam a gente a não desistir.

15.2. A biblioteca de sons do Scratch

Localizada no campo "Sons", ao lado de "Fantasias", nós temos um menu muito similar ao que já vimos nas próprias fantasias e onde já explicamos de forma breve anteriormente, mas temos mais algumas boas informações para você! Assim como diversos atores e cenários, o Scratch possui muitos sons diferentes que podem ser usados em cenários e atores, além também de diversas ferramentas de edição, o que vai facilitar muito a sua vida, pode confiar!

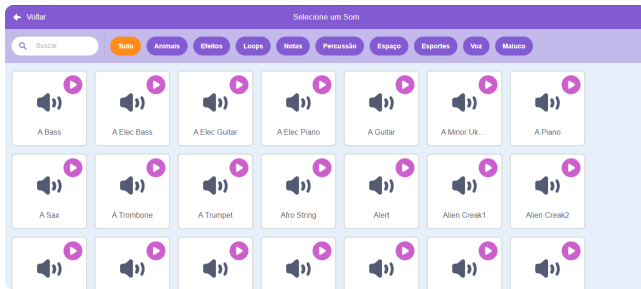


Vamos dar destaque agora para um botão muito especial, o botão "Selecione um Som"! É

clitando nele onde podemos adicionar mais áudios e acessar a biblioteca de sons do Scratch.



Existem muitas opções que aparecem quando se passa o mouse por este botão, o que você acha de testar elas? Faça seus experimentos!



Entre muitos campos, temos efeitos, músicas, vozes e coisas muiiiito aleatórias. Faça seus testes!

15.3. Exercícios

O exercício desse capítulo será bem simples, porém muito útil, nós vamos continuar editando o nosso jogo de onde paramos, lá na Prática!

1. Estando com o ator Sprite1 selecionado, vamos até Eventos!

2. Nos eventos o que iremos fazer será posicionar logo abaixo do conjunto de blocos que fazem o ator nadar para cima e para baixo, o bloco "quando 'Ir' for clicado".

3. Agora vamos até a categoria "Controle", lá nós iremos arrastar até abaixo do último bloco que posicionamos o bloco "sempre".

4. E então, dentro ainda da categoria "Controle", posicionaremos dentro do bloco "sempre", o bloco "espere até que".

5. E então, vamos ir até "Sensores", lá nós posicionaremos dentro do teste do bloco "espere até que" o bloco "tocando em 'ponteiro do mouse?'". Também mudaremos o valor "ponteiro do mouse" para "borda".

8. E então vamos até a categoria "Som", dela arrastaremos até abaixo do último bloco que

posicionamos dentro do laço "sempre", o bloco "toque o som 'Meow' até o fim".

9. E então, vamos ao campo "Sons", que fica ao lado de "Fantasias". Lá nós vamos clicar no botão "Selecione um Som".

10. Aqui você pode selecionar o som que você quiser! Mas nós iremos escolher o som "Bubbles", você também pode pesquisar por ele, digitando "Bubbles" na barra de pesquisa superior. Nem precisa apertar Enter!

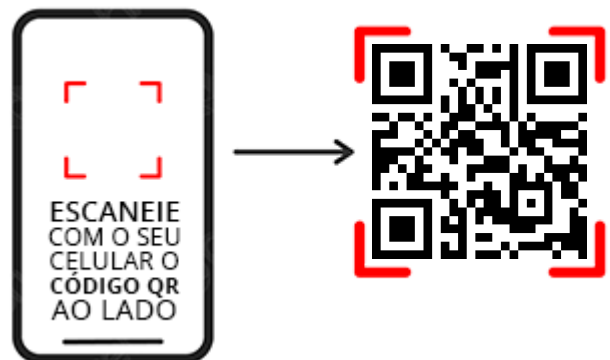
11. Depois de clicar no som, ele será adicionado a sua área de trabalho no campo "Sons", nós iremos cortar metade dele, para fazer isso é bem simples, basta clicar mais ou menos no meio do grande quadrado rosa cheio de ondinhas, segurar o clique, e arrastar até o extremo do lado direito do quadrado, você vai ver que a metade dele será selecionada!

12. Depois disso, basta clicar no botão em forma de tesoura escrito "Apagar".

13. Depois disso, voltamos para o campo do "Código" e alteramos o valor do bloco "toque o som 'Meow' até o fim" para "Bubbles".

E está feitiinho da silva! Apenas temos que jogar! Hehehe...

Você pode fazer isso acessando o QR code abaixo ou copiando e colando o link na barra de pesquisas do seu navegador!



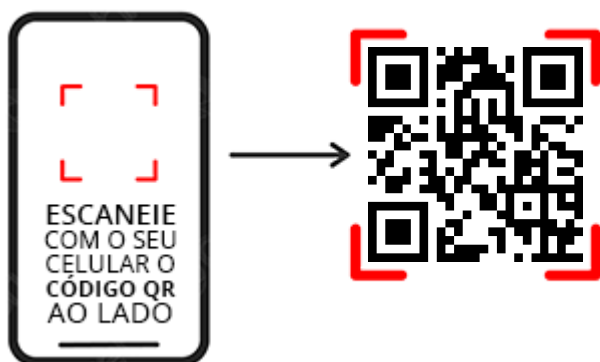
<https://aposti.la/5lexv>

Olá e bem-vindo(a) à última apostila do curso de Programação Kids! Vamos concluir o projeto final, o jogo que estamos produzindo há várias aulas. Hoje, adicionaremos um peixe venenoso, um temporizador e um contador de vidas. No final, você poderá baixar o arquivo do projeto completo para editar, modificar ou criar um novo do jeito que quiser!



16.1. Exercício

Então, como foi falado acima, no exercício final (que é hoje), vamos adicionar um contador de vidas, um temporizador para sabermos quanto tempo estamos jogando e um peixe venenoso, que ao capturarmos, tomaremos um dano, sendo assim, perdendo uma vida!



<https://apostila.la/jjbw4>

Então, copie o link, escaneie o QR code ou acesse os arquivos auxiliares da aula para ter acesso à continuação do que fizemos na prática da última aula, utilize então a codificação para fazer o exercício final!

1. Vamos lá: primeiramente certifique-se que o ator selecionado é o Scratch, após isso vamos precisar de novas variáveis, então vá para a categoria "Variáveis" e crie duas novas variáveis, uma com o nome "Vidas" e outra com o nome "Tempo".

2. Começando a codificação de fato, vá até a categoria "Eventos", como de costume, e arraste o bloco de clique na bandeirinha até o local que você preferir.

3. Vá até as variáveis e adicione o bloco "mude". Embaixo do bloco "quando 'ir' for clicado", mude a variável desse bloco para "Vidas" e troque o valor de '0' para '3'.

4. Agora preste atenção: vá para "Controle" e adicione o bloco "sempre". Dentro dele, coloque o bloco "se então". No bloco "se então", adicione o bloco "pare" e mude seu valor para "outros scripts no ator". Em seguida, adicione outro bloco "pare" com o valor "todos".

5. Agora, vá para a categoria "Operadores" e adicione o bloco "=" dentro do bloco "se então". Vá para a categoria "Variáveis" e adicione a variável "Vidas" para dentro do espaço vazio do bloco "=", e mude o valor de '50' do bloco "=" para o número "0".

6. Vá para a aba 'Sons', 'Efeitos', procure e adicione o som 'Lose'. Agora, volte para a aba 'Código' e procure o bloco "toque o som até o fim", na categoria "Som", após isso, adicione esse bloco no meio dos dois blocos "pare", e então, mude o valor dele para "Lose". Pronto, o contador de vidas foi finalizado!

7. Agora vamos criar o código para contar o tempo que estamos jogando, para isso, adicione um bloco "quando 'ir' for clicado" onde você quiser, pode ser logo abaixo do código do contador de vidas que acabamos de fazer. Vá para a categoria "Controle" e arraste os blocos "sempre" e após isso arraste o bloco "espere" para dentro do bloco "sempre".

8. Vá para a categoria "Variáveis", e após isso, adicione o bloco "mude" logo abaixo do bloco de clicar na bandeirinha, e então, adicione o bloco "adicione" para baixo do bloco "espere 1 seg". Mude os valores desses dois blocos da categoria "Variáveis" para a variável "Tempo". E nosso contador foi finalizado!

9. Agora vamos criar o peixe venenoso, para isso, duplique o peixe vermelho, clique na aba "Fantasias" após ter selecionado o peixe novo, agora, pode colorir ele do jeito que você quiser ou, de roxo, que é a cor que ele estará na versão final do projeto.

10. Após colorir ele, vá para a aba de 'Sons' e procure pelo som "Bell Toll" que também está na categoria 'Efeitos', selecione e em seguida vá para a aba "Código". Procure pelo bloco "toque som" e mude o valor dele para "Bell Toll".

11. Agora, no bloco "adicione" que está abaixo do bloco "toque som 'Bell Toll'", coloque o valor '-1' nele e mude a variável, para a variável "Vidas".

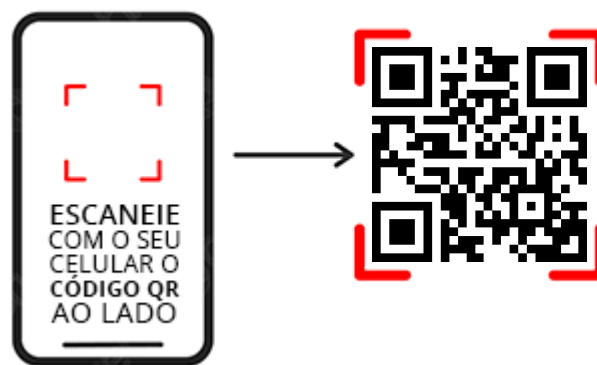
12. Está quase completo, para o último passo, clique no ator "Sprite1", o Scratch, mais precisamente. Agora, preste bastante atenção, duplique o bloco "sempre" que tem o bloco "se então", ele é o bloco que foi colocado antes, durante a criação do contador de vidas, vamos precisar dele e de alguns que estão dentro dele para colocar nos peixes!

13. Após duplicá-lo, arraste esse conjunto todo para dentro de todos os peixes, uma vez para cada, e então, esse conjunto que você duplicou para enviar para os peixes você pode excluir.

14. Agora, vá em cada peixe que você enviou o código duplicado e faça os seguintes passos: Adicione o bloco da bandeirinha, o "quando 'ir' for clicado"; Exclua os blocos "toque o som" e o bloco "pare 'todos'".

Senhoras e senhores o nosso projeto está finalizado! Novamente te agradecemos por ter nos acompanhado nessa jornada da programação. Esperamos que o que lhe foi ensinado sirva como motivação para aprender mais e mais sobre essa área!

Não esqueça: você pode editar, modificar ou fazer o que quiser com o projeto, basta acessar o link do QR code ou verificar os arquivos auxiliares dessa aula para encontrar o arquivo do projeto, então, basta abri-lo na plataforma Scratch ou aplicativo! Até nossa próxima aventura!



<https://aposti.la/gcekt>

anotações
