



conecta makers



Material organizado por: Giovana Melos Borsoi, Maria Eduarda Rech,
Dr. Leonardo Poloni, Dra. Kelen Berra de Mello



3- ATIVIDADE MÃO NA MASSA: CRIANDO JOGOS COM SCRATCH

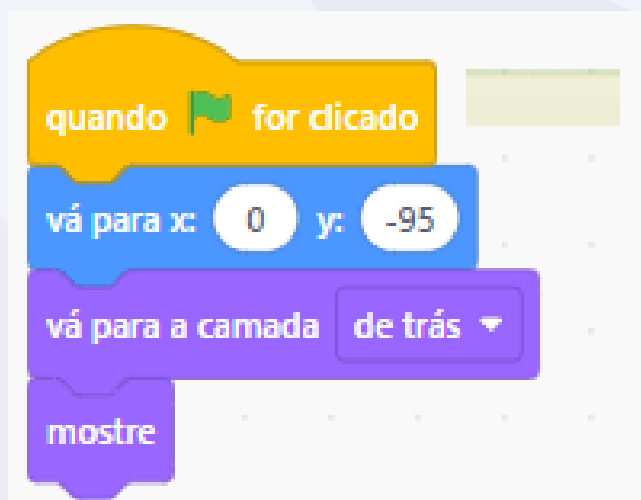
Você já se perguntou como são criados os jogos de celular? Vamos desenvolver uma adaptação do famoso jogo “Flappy Bird”

1. Escolha um cenário de sua escolha. Você pode baixar um cenário ou criar o seu próprio.
2. Escolha os personagens que você gostaria de animar.
3. Posicione todos os personagens aonde você gostaria que eles estivessem ao iniciar a sua programação.
4. Vamos sempre pensar na sequência do jogo, o que queremos que aconteça ao iniciá-lo?

Como queremos que ele inicie?

Neste caso queremos que ao clicar na bandeira verde, apareça a tela inicial do jogo com o botão START. Para isso vamos programar todos os atores desta tela (chão, botão start, nome do jogo) para que fiquem aparentes, enquanto os outros atores devem ficar escondidos ao clicar na bandeira verde.

No ator “chão” iremos apenas adicionar o bloco de evento para iniciar a execução da programação, o bloco de posição que faz com que o ator se direcione até onde queremos e os blocos de aparência, para que o ator vá para camada de trás e não tampe os outros atores e o mostre para que ele fique visível.



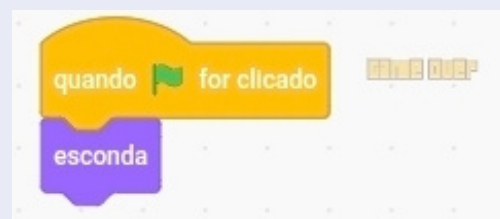
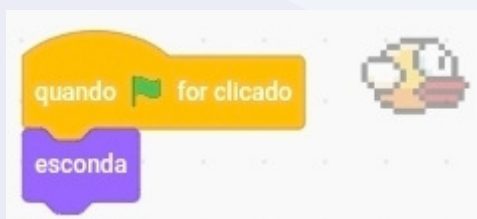
Para o ator “START” queremos que ele vá para a cena quando a bandeira verde for clicada e quando for clicado, ele transmita uma mensagem "avisando" que o jogo começou. Para isso usamos os blocos de evento “Quando este ator for clicado” e “transmita” a mensagem “iniciar jogo” e depois usamos o blocos de aparência “esconder”.



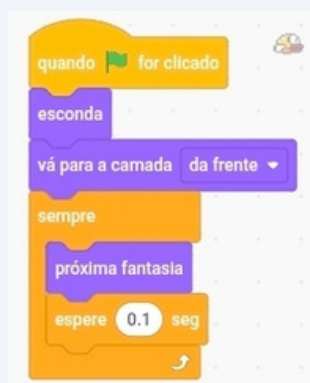
Para o ator “nome do jogo” queremos que ele vá para a cena quando a bandeira verde for clicada, e se esconda ao iniciar o jogo. Para isso usamos os blocos de evento para dizer quando vai acontecer e os de aparência para dizer se queremos que mostre ou esconda.



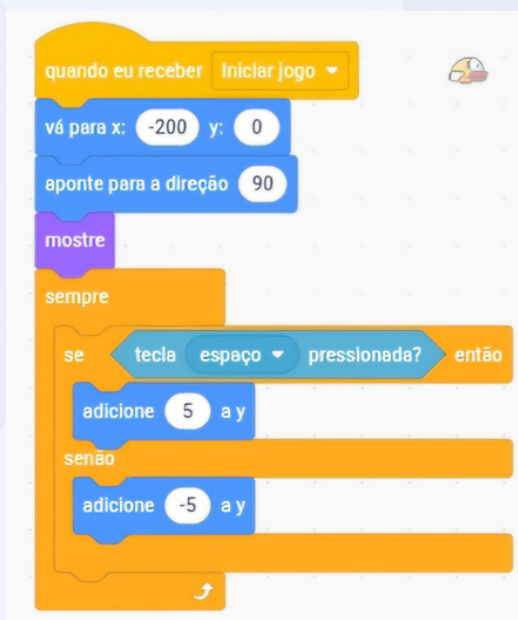
Agora vamos programar o restante dos atores para que quando a bandeira verde for clicada eles se escondam. Para isso usamos o bloco de evento e o de aparência



Neste momento, vamos planejar quais ações devem acontecer antes de o botão "START" ser pressionado. Ou seja, antes de começar o jogo, tem mais alguma coisa que precisa acontecer? No nosso caso, temos que ajustar a posição dos obstáculos para que eles iniciem "fora da tela". Para isso vamos usar o bloco de movimento para que ele vá até a posição que queremos. Também vamos programar o passarinho para que ele vá para a camada da frente e troque a fantasia diversas vezes para que dê a impressão de estar voando e para isso vamos utilizar os blocos de aparência e de controle.



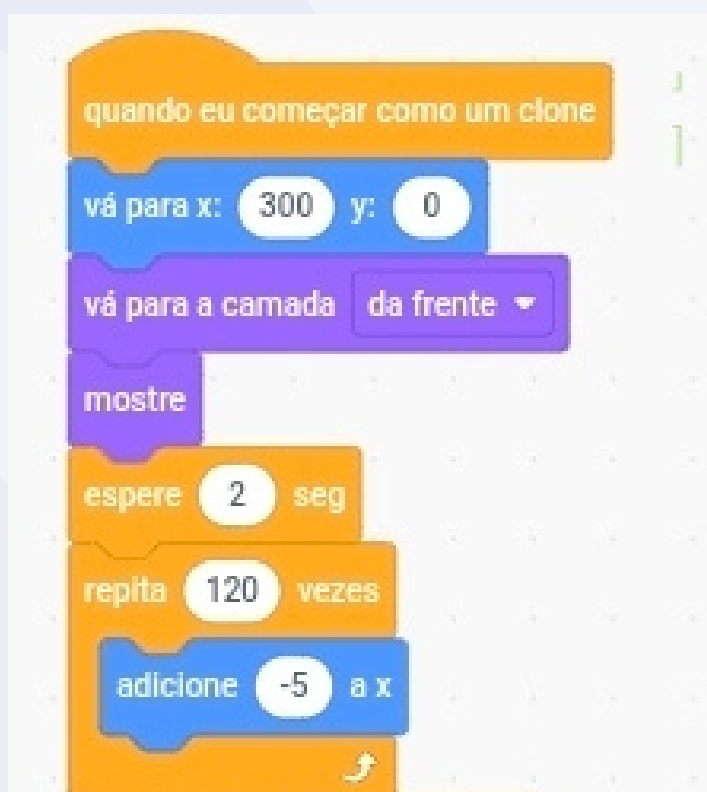
Agora vamos pensar o que queremos que aconteça quando o botão "START" for pressionado, que seria quando a mensagem "começar jogo" for transmitida, logo o bloco de evento a ser usado é "quando eu receber 'iniciar jogo'". Vamos começar com o passarinho, o que queremos que aconteça com ele quando o jogo iniciar? Queremos que ele fique sempre com a mesma posição de x e altere apenas os valores de y, ou seja, ele não anda na horizontal, apenas sobe. Além disso, queremos que ele fique sempre caindo, e ao clicar na tecla espaço ele suba, fazendo assim o movimento inicial do jogo. Para isso, vamos utilizar o bloco de evento para dizer quando vai acontecer, os blocos de movimento para que ele vá para onde queremos e aponte para a direção que queremos, o bloco de aparência para que ele mostre e os blocos de controle, sensor e movimento para que ele faça o movimento de subir e descer.



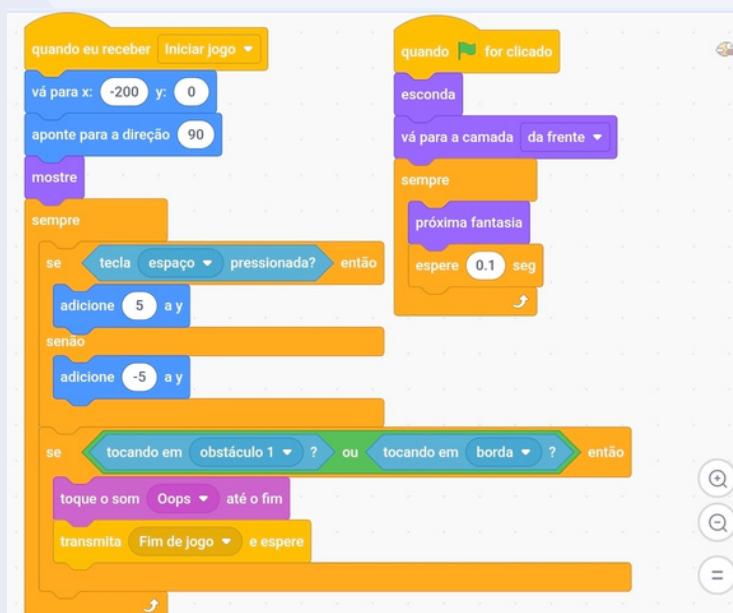
Para programar os obstáculos, precisamos que eles deem a impressão de estar deslizando para a esquerda até "sair da tela" novamente. Para isso, precisamos utilizar um clone e trocar as fantasias deste clone para que a abertura dos canos esteja em diferentes alturas, também utilizaremos um bloco de espere para que os clones não sejam criados todos ao mesmo tempo.



Na programação do clone, vamos precisar que ele vá para "fora da tela", que vá para a camada da frente e apareça, que espere um tempo para que os clones não se sobreponham, e vá para a esquerda até "sair da tela" novamente. Para isso, iremos utilizar os blocos de controle, movimento e aparência.



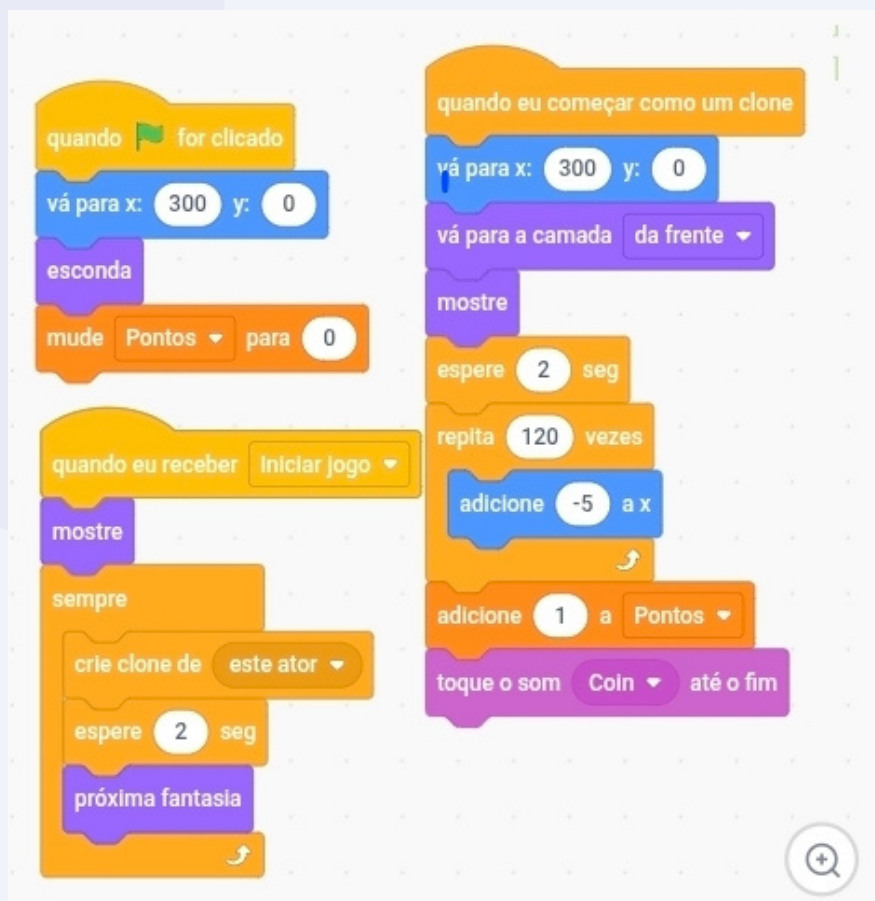
Neste momento o nosso jogo está funcionando, mas falta uma parte importante, o fim dele. O jogo "acaba" quando o jogador deixar o passarinho cair muito ou bater em um obstáculo. Então, vamos programar o passarinho para que quando ele estiver tocando nos obstáculos ou na borda, transmita uma mensagem de "Fim de jogo". Para isso, utilizaremos os blocos de controle, operadores, sensores e evento, e caso você queira, pode adicionar um som, basta selecioná-lo na aba de sons e adicionar utilizando um bloco da categoria sons.



Vamos programar o ator "Fim de jogo" para que quando o passarinho transmitir uma mensagem de fim de jogo, apareça para o jogador uma tela diferente, escrito 'game over'. Para isso, vamos utilizar os blocos de evento, aparência e controle. Utilizaremos o bloco "pare todos" para indicar que queremos que todas as programações de todos os atores neste momento sejam pausadas, então os atores irão parar no lugar em que estão



Para programar a pontuação, iremos criar uma nova variável e denominá-la 'pontos', então iremos escolher algum dos atores para realizar a programação destes pontos, no nosso caso, escolhemos os obstáculos já que é quando eles chegam ao fim da tela que o jogador marca um ponto. Para isso, iremos colocar um bloco de variáveis no evento inicial, para que sempre ao iniciar o jogo os pontos estejam zerados. Também iremos utilizar outro bloco de variável, mas desta vez na programação do clone, para que sempre que o clone chegar no extremo esquerdo da tela, marque um ponto e, caso você queira pode também adicionar um bloco de som nesta parte.



AGORA É COM VOCÊS!

Desafio: Agora que você já aprendeu os principais blocos do Scratch e suas funcionalidades, tente criar um botão de pause, para que o jogador possa pausar o jogo sem interferir no desempenho adquirido até o momento.

REFERÊNCIAS

BRACKMANN, C. P. Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. Tese (Doutorado em Informática da Educação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017. Disponível em: <https://lume.ufrgs.br/handle/10183/172208> .

BRACKMANN. Computacional: Educação em Computação. 2026. Disponível em: <https://www.computacional.com.br/> .

BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília, 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/>.

LIUKAS, L. Hello Ruby: adventures in coding. [S.l.]: Macmillan, 2015.

PAPERT, Seymour. Logo: computadores e educação. São Paulo: Brasiliense, 1985.

Pensamento Computacional e Ciência da Computação Desplugada. CS Unplugged, 2024. Disponível em: <https://www.csunplugged.org/en/computational-thinking/>.

RESNICK, M. Jardim de infância para a vida toda: Por uma aprendizagem criativa e colaborativa. Porto Alegre: Penso. 2020.

WING, J. Computational Thinking. Tradução de Cleverson Sebastião dos Anjos. PENSAMENTO COMPUTACIONAL – Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. Revista Brasileira de Ensino de Ciência e Tecnologia, v. 9, n. 2, p. 1-10, mai./ago. 2016. Disponível em: <https://revistas.utfpr.edu.br/rbect/article/view/4711> .