MOONHACK 2019

SCRATCH PROJECT

PORTUGUESE

Introdução

Em julho de 1969, Neil Armstrong deu os primeiros passos na lua. Nós então celebramos o 50º aniversário desta conquista incrível, nós temos a chance de refletir na dificuldade e na audácia que foi cada parte das missões lunares. Neste projeto, iremos usar o Scratch para fazer um um jogo de um Buggy Lunar.

O que iremos precisar:

Hardware
Um computador capaz de rodar Scratch 3

Software
Scratch 3 (pode ser online ou offline)

Passo 1: Configurar a superfice

Lista de atividades

- [ ] Abra o projeto inicial em https://scratch.mit.edu/projects/293834812/editor/

- [ ] Primeiro, nós iremos configurar a imagem da superfície para ela mover-se enquanto o buggy ficar parado, simulando movimento. Clicar na imagem da superfície e adicionar o código a seguir:

Este código irá ser executado durante o loop `sempre`

- [ ] Agora iremos adicionar junto ao nosso loop `sempre` um código para mover nossa superfície:

Nós iremos usar a variável `scrollx` para acompanhar quão longe nosso buggy se moveu, e mover a superfície de junto.

Passo 2: Dirigir o buggy

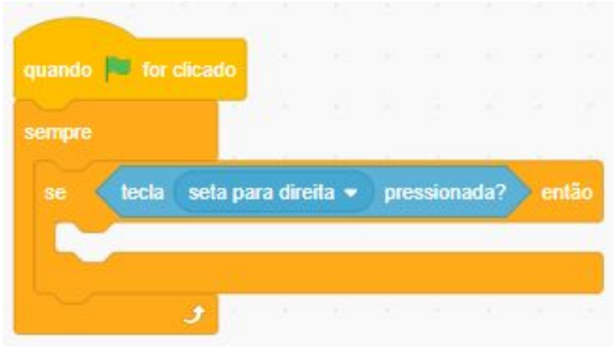
Se você rodar seu programa agora, nada muito interessante irá acontecer. Nós estamos dizendo a superfície para mover-se com a variável `scrollx` mas não estamos mudando a variável `scrollx`.

Lista de atividades

- [ ] Clicar na imagem do Buggy e adicionar o seguinte código:

Juntamente com o loop, nós adicionamos uma estrutura de `Se..Então`. O código dentro desse bloco irá apenas rodar se a condição do `Se..Então` for satisfeita.

- [ ] Adicionar bloco `tecla...pressionada?` como nossa condição no `Se..Então`. Mude a tecla de `espaço` para `seta para direita` clicando na setinha da caixa de seleção próxima a opção `espaço`.



- [ ] Adicione o código para mudar `scrollx` para -2 para fazer a superfície mover-se quando a tecla `seta para direita` for pressionada:

Rode o programa. A superfície deve mover-se para esquerda, o que vai parecer que o Buggy está se movendo para a direita. O que aconteceu quando você chegar ao fim da imagem da superfície? Se você tentar iniciar novamente, a superfície inicia no mesmo lugar. Um jogo não seria muito bom se você só puder jogá-lo uma vez!

- [ ] Nós iremos adicionar um código para voltar com a Superfície de volta ao centro quando clicarmos na `bandeira verde`.

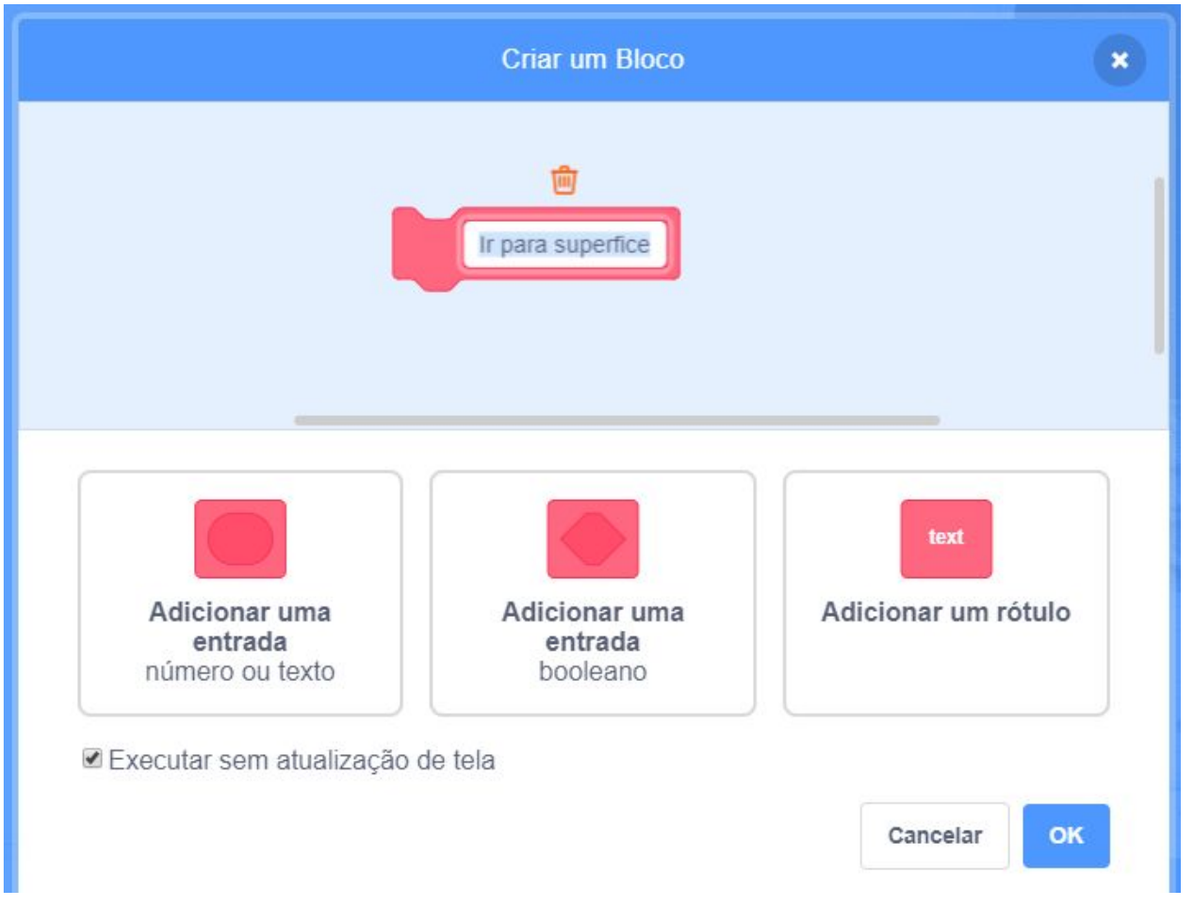
Desafio avançado: Andando de marcha ré

Atualmente, o buggy apenas vai para frente, você poderia criar um código para permitir que ele vá para trás também?

Passo 3: Fazer o buggy ir para superfície

Lista de atividades

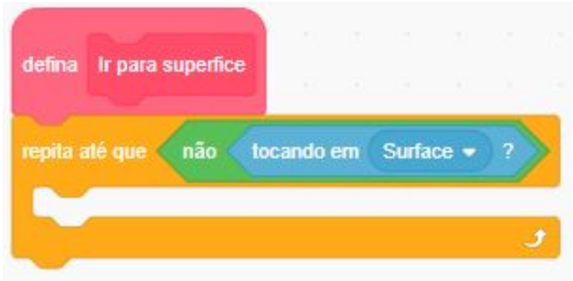
- [ ] Nós iremos criar nosso próprio bloco de código que irá fazer com que o buggy sempre vai para o topo da imagem da superfície. Vá para **`meus blocos`** e clique em **`fazer um bloco`**. Nomeie o seu bloco como `ir para superfície` e na opções clique em **`executar sem atualização de tela`**.



Normalmente, quando nós temos um loop, ele irá rodar uma vez dentro do loop a cada quadro do jogo. Marcando **`executar sem atualização de tela`** ele irá rodar o bloco inteiro sem necessidade de aguardar o próximo frame.

- [ ] Quando você pressionar OK um bloco `ir para superfície` vai aparecer na sua tela. Adicione um bloco `repita até que`, como abaixo:

- [ ] Nós queremos repetir o código dentro desse bloco enquanto não estivermos tocando na superfície.



- [ ] Toda vez que estivermos no loop, queremos que ele vá para cima 1 pixel:

Isso irá empurrar o buggy para cima sempre que ele estiver tocando a imagem da superfície, até que não esteja mais o tocando.

- [ ] Nós terminamos de escrever nosso bloco `ir para superfície`, mas não estamos usando ele ainda. Precisamos adicionar no loop principal.

Ao executar o jogo agora. Nosso Buggy irá navegar bem nas subidas, mas não vai conseguir acompanhar as descidas.

- [ ] Para fazermos o Buggy acompanhar as descidas, precisamos adicionar o efeito de gravidade a ele. Adicione um bloco `adicione a y` para que o Buggy lentamente caia se ele estiver no ar:

Passo 4: Mais superfície

Até agora, não podemos ir muito longe na Lua até que caímos em um abismo. Vamos fazer com que a experiência lunar seja melhor aproveitada.

Lista de atividades

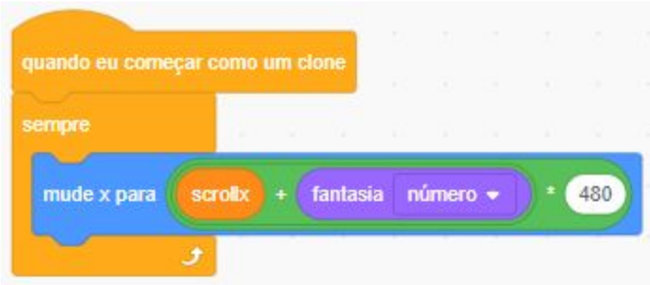
- [ ] Clicar na imagem da superfície e separe o bloco `quando a bandeira verde for clicada` do resto do código:

- [ ] Abaixo do bloco `quando a bandeira verde for clicada` adicione código para configurar as condições iniciais:

- [ ] Adicionar um loop que irá criar um clone de cada fantasia da imagem:

- [ ] No código que você separou adicione `quando eu como começar como clone`:

- [ ] Porque existem muitas fantasias que podem compor a superfície da lua, nós precisamos movê-las ao longo da tela com nossa variável `scrollx`, mas cada fantasia deve estar afastada 480 pixels da anterior (que é a largura da fantasia):



- [ ] Se executarmos o jogo agora, a superfície não irá aparecer, porque nós dissemos anteriormente para escondê-la. Não queremos exibir todas as imagens porque o Scratch não deixa nossas imagens ir todas para fora da tela. Ao invés disso, nós podemos dizer ao Scratch que se a coordenada X do estiver como planejamos, então mostre, do contrário, esconda.

Dica: Se você não quiser reescrever uma parte do código, você poderá duplicá-lo clicando com o botão direito do mouse e selecionando Duplicar:

- [ ] Se você rodar seu jogo agora, nossa primeira imagem de superfície irá iniciar fora da tela à esquerda, você pode consertar isso clicando no Buggy e configurando o valor inicial da variável `scrollx` para -480:

Passo 5: Pulando com o Buggy

Nós queremos fazer com que nosso Buggy evite os obstáculos!

Lista de atividades

- [ ] Adicione o código abaixo para fazer com que nosso Buggy pule 5 pixels quando a tecla de espaço for pressionada.

Passo 6: Pulando os abismos e chegando ao fim

Ao longo da nossa jornada, você pode ter percebido alguns abismos. Nós não queremos cair neles, não é mesmo? Então vamos adicionar algum código para que possamos pulá-los e chegar ao final.

Lista de atividades

- [ ] Primeiramente, nós precisamos dizer ao jogo que perdemos se cairmos no abismo. Nesse caso, iremos usar o bloco `tocando na borda` para checar se estamos tocando na borda, e se acontecer, é fim do jogo. No fim do nosso loop `sempre` do nosso Buggy adicione o código a seguir:

Execute o jogo agora. Se você cair em um abismo, será o fim do jogo.

- [ ] Nós agora temos um modo de perder o jogo, precisamos agora programar a condição de vitória. Adicione o código a seguir ao fim do loop `sempre` do Buggy quando atingirmos o objetivo:

Tenha ação pois por estarmos usando valores negativos para a variável `scrollx`, precisamos usar o bloco < (menor que).

Desafio: Adicionar um cronometro

É muito legal poder comparar seu melhor tempo com seus amigos. Você pode adicionar um cronômetro para medir quanto tempo levou para ganhar o jogo?

Desafio avançado: Pulo melhorado

Agora, quando pulamos, não há nada que pare o buggy de pular ou que mude sua direção ao longo do pulo. Na lua, você não pode fazer isso! Você pode atualizar seu jogo para que o buggy apenas pule uma vez, e não pode pular novamente ou mudar a direção até que ele caia no chão.

Dica: Com uma variável dentro do REPITA ATÉ QUE de seu bloco Ir para superfície irá ajudar a acompanhar se está ou não tocando no chão.