Caro educador!

Este guia é o produto educacional de uma pesquisa do Programa de Pós-Graduação em Ensino de Ciências e Matemática que tem como objetivo o desenvolvimento de habilidades do Pensamento Computacional para estudantes do 4º a 8º ano do Ensino Fundamental.

Abaixo estão descritos os encontros a serem realizados. No total são 8 encontros de aproximadamente uma hora e meia cada um, passível de adaptação.

Para o desenvolvimento das atividades é necessário computadores simples com acesso a internet (um computador para cada grupo de 4 alunos é suficiente) ou com o software Scratch instalado. Também será utilizado um projetor.

Para mais informações sobre o software, visite o website https://scratch.mit.edu/, que conta com o próprio software para download ou para uso on-line, além de tutoriais e guias.

Encontro 1 – Introdução: Construindo nosso primeiro programa

Ao iniciar uma sequência didática é importante apresentar aos estudantes a proposta deste, o objetivo, o método a ser utilizado e, especialmente se tratando de atividades que envolvem um software específico, é necessário que ele seja apresentado aos estudantes de maneira que não se sintam intimidados.

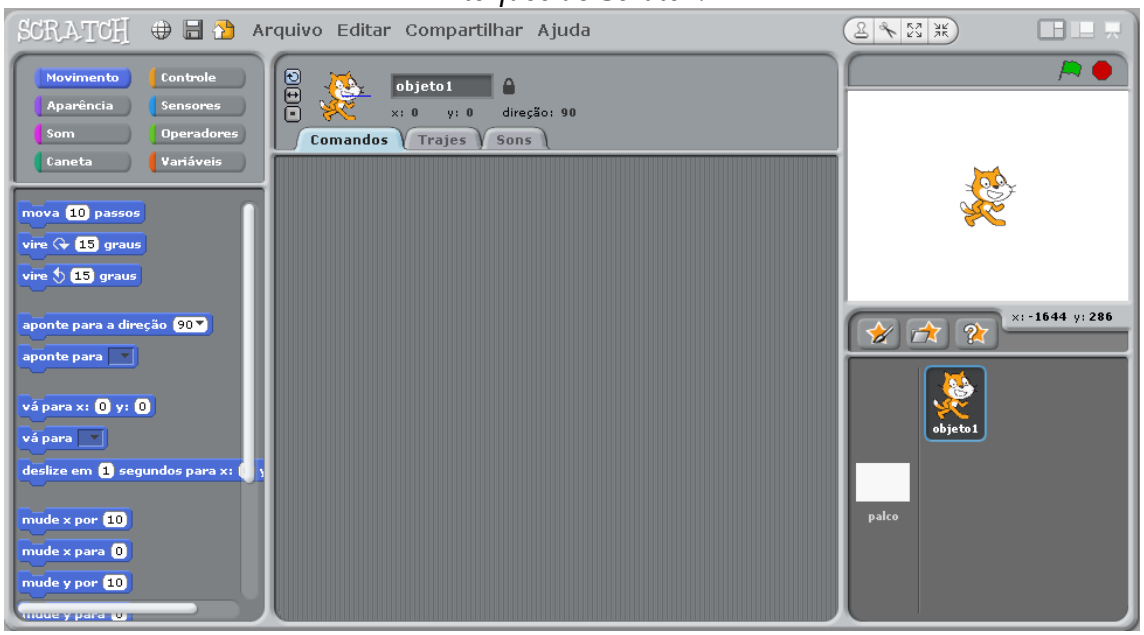
Descrição da Atividade

Aos estudantes, explicar o funcionamento dos encontros: quando acontecerão, como acontecerão, por que estarão realizando essas atividades (objetivo) e o que é esperado deles. Da mesma maneira, os estudantes serão questionados em relação a suas expectativas.

Introdução ao *Scratch*

Utilizando um projetor, apresentar aos estudantes a interface do software *Scratch.*

Interface do Scratch.



Deve ser feita uma apresentação breve sobre como funciona o software e a linguagem de programação em blocos, específica deste. Não se espera que os estudantes aprendam todos os tópicos comentados, esse aprendizado virá conforme eles agem sobre o objeto.

Definindo o Problema

Agora é a hora de construir o primeiro programa destes estudantes (daqueles que nunca tiveram experiências em programação), que será uma animação simples com a personagem gato do *Scratch*.

Este programa será feito por todos, com acompanhamento do professor, na tela que está sendo projetada. Devem, então, definir um roteiro para a animação.

A animação deve ser algo simples, que pode ser adaptada conforme sugestões dos estudantes. Pode ser, por exemplo, animar a personagem para dar alguns passos para frente e depois dizer "Olá, meu nome é *Scratch*!".

É importante que o professor esteja aberto a sugestões dos estudantes, para customizar[1] o programa e fazer com que este tenha significado para toda a turma. Neste texto, iremos tratar a animação como descrito acima.

Durante a construção do programa, o professor deve sempre encorajar os estudantes a utilizarem habilidades do Pensamento Computacional (vamos testar?, como podemos separar esta animação em partes?, algo está errado, será que conseguimos consertar?). Vamos utilizar essas habilidades, que queremos desenvolver, para a resolução deste problema.

O professor irá encorajar o estudante a **decompor o problema** em partes menores. Neste caso, podemos dividir a animação em duas partes: movimentar a personagem e a fazer falar. Vamos analisar cada parte separadamente.

Parte 1: Movimentando o Objeto

O professor pode pedir aos estudantes que procurem nos blocos de programação, e encontrem algum que possa fazer com que nosso objeto se movimente. Intuitivamente, os estudantes serão levados a encontrar o bloco de mover.

Bloco de movimento.

Na sequência, irão **realizar o teste**: dê dois cliques no bloco em questão. Eles deverão observar que isso movimentou a personagem no sentido que ela está virada.

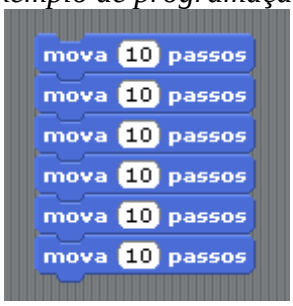
[1] Customizar é modificar, adaptar ou personalizar algo de modo a adequá-lo às suas necessidades.

Podem, então, realizar o teste para ver o que acontece quando aumentam o número de passos.

Então, para movimentar o objeto, podem utilizar este bloco. Deve ser observado que ao mover o objeto desta forma, ele desaparece de onde estava e aparece na posição mais adiante, dependendo do valor estabelecido. Portanto, apenas este bloco não é suficiente para fazer a personagem caminhar.

O passo mais natural a seguir é o de tentar colocar mais blocos de movimento. É esperado que os estudantes experimentem algumas formas diferentes, e cabe ao professor lidar com elas. Podem testar, todos juntos, algumas sugestões, como o exemplos abaixo.

Exemplo de programação.



Deparamo-nos, então, com outro problema: é preciso que a personagem dê uma pausa antes de dar o próximo passo. Para isso, desafiamos os estudantes a encontrar uma solução.

O bloco de espera pode ser uma sugestão dos estudantes. Vamos então, tentar colocar um bloco de espera entre os passos.

Programação com bloco de espera.

Agora a personagem parece que está caminhando. Completamos o nosso programa. Podemos também ajustar o tempo de espera.

Ajustes para a programação.

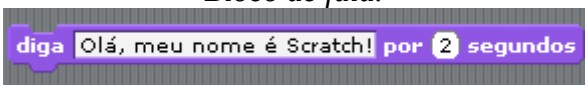
O professor pode ajudar a turma a descobrir que este processo pode ser sintetizado com o bloco de repetição.

Programação simplificada.

Parte 2: Fazendo a Personagem Falar

Para realizar a segunda parte do nosso problema, a turma pode explorar o software para encontrar uma solução. Uma delas pode ser o bloco a seguir.

Bloco de fala.

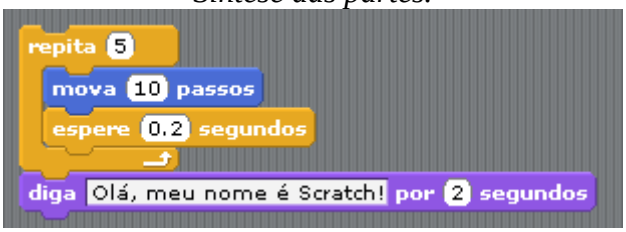


Alternativamente, podem ser utilizados os blocos de som, com gravações. Tudo depende de como a turma encarar o desafio.

Unindo os módulos (síntese)

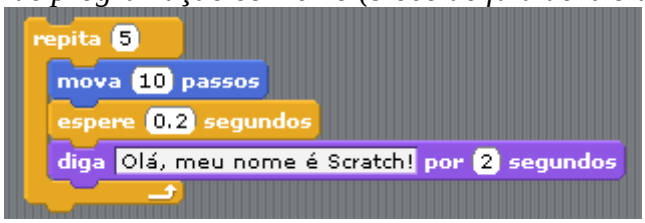
Fazendo a síntese das partes do problema, conseguimos construir o programa propost.

Síntese das partes.



A realização dessa síntese deve ser discutida também. O conceito de *loop* ainda não foi formalmente apresentado para a turma, e o uso errado do *loop* pode acontecer, como por exemplo colocando o bloco de fala junto com a parte de movimento.

Exemplo de programação com erro (bloco de fala dentro do loop).



Recursos Extras

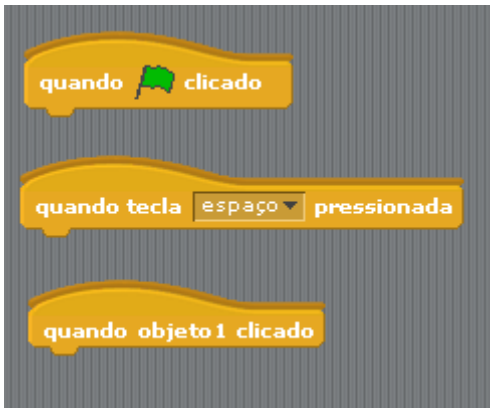
De acordo com o ritmo da turma e a atenção que o professor pretende dar a cada detalhe, é possível que sobre tempo para explorar outras funções do programa.

Uma das funções que pode ser explorada são os trajes, para fazer o movimento parecer mais natural.

Função "trajes".

É possível também explorar os blocos de comando para iniciar o programa sem precisar dar dois cliques no programa.

Blocos de comando.



Exploração Livre

Agora que a turma programou uma pequena animação, devem ser encorajados a modificar esta, ou até mesmo criar uma diferente, com os recursos que quiserem. É o momento de explorarem os recursos do software e sua capacidade intuitiva de programação.

Apresentação

Finalmente, os estudantes irão apresentar seus programas para os colegas. Serão encorajados a explicar o que utilizaram de diferente, se encontraram muitas dificuldades e o que acharam de ser programadores.

Avaliação

A avaliação do encontro será feita de duas maneiras:

a) *Das habilidades do Pensamento Computacional utilizadas*: nos programas apresentados, será feito um levantamento das habilidades e conceitos do Pensamento Computacional que se fizeram presentes. Além disso, durante o encontro essa análise deverá ser feita em todos os momentos pelo professor.

b) *Da validade do encontro para o estudante*: através de um breve relatório, será feito um levantamento de dados sobre os estudantes em relação a suas experiências prévias com programação e primeiras impressões.

Nome	Idade	Turma

Algum dos integrantes da equipe já conhecia o Scratch? Se sim, quem?

O que vocês conseguiram fazer com o Scratch? Como conseguiram isso?

Arraste para o quadrado o *emoji* que representa o que você achou das atividades de hoje:




Encontro 2 – Let's Dance

O Pensamento Computacional é uma habilidade que está presente também quando não estamos programando. Esta atividade inicia com o desafio de "programar uma pessoa", e segue com a transposição deste desafio para a interface do *Scratch*.

A ideia de programar uma pessoa é simplesmente estabelecer uma sequência de passos para resolver um problema. Nesse caso, o problema é ensinar uma dança a uma pessoa sem utilizar elementos visuais, apenas explicando os passos.

Descrição da Atividade

Separados em duplas, será designado a cada estudante um papel: programador ou programa. Cada dupla deverá ser composta de um programador e um programa.

Vamos dançar!

Aos programadores será mostrado um vídeo de dança. Ele deverá passar ao programa as instruções através da fala. Será proibido ao programador utilizar gestos ou mostrar a dança de alguma maneira que não por comandos verbais.

O outro integrante, o programa, deve então realizar a dança conforme entender, e depois irá comparar seu resultado com a dança original.

Será feito uma pequena discussão sobre a importância da clareza quando estamos programando e nas tarefas do cotidiano.

Festa do *Scratch*!

Ainda separados em duplas, os estudantes irão programar uma "festa do *Scratch*". Para isso, cada dupla seguirá o tutorial[2] e, em seguida, será encorajada a personalizar seu projeto adicionando objetos, alterando a dança, o pano de fundo e explorando os sons. O desafio é fazer a animação de uma festa de dança.

Cada grupo irá apresentar sua animação para a turma. Enquanto isso, devem preencher fichas de *feedback*, que serão entregues para a avaliação por pares.

[2] https://scratch.mit.edu/projects/editor/?tip_bar=getStarted

Ficha de feedback.

FICHA DE FEEDBACK	O que você acha que poderia ter sido melhor?	Você achou algo confuso, que poderia ter sido feito de outra forma?	O que você achou legal e que funciona bem no projeto?
DE: grupo ___			
PARA: grupo ___			

Encontro 3 – *Debug it*!

Uma das habilidades do Pensamento Computacional é a habilidade de depurar: encontrar erros e consertá-los. Com este objetivo, neste encontro serão propostos desafios na forma de programas que estão com erros. Cabe a cada grupo encontrar os erros e consertá-los.

Descrição da Atividade

Separados em grupos, os estudantes irão receber os seguintes programas com seus respectivos erros (*bugs*). Eles devem encontrar a parte do programa que está causando este erro e consertá-la.

Será solicitado que eles registrem em um pequeno formulário qual era o erro do programa e como ele foi corrigido.

No final da aula, serão discutidas as soluções.

Desafio 1: Gobo está Parado!

Neste programa, tanto o gato do *Scratch* quanto o Gobo (personagem amarelo) deveriam dançar quando a bandeira verde é clicada, mas isso não acontece. Cabe a cada equipe encontrar o erro e consertá-lo, do seu jeito.

Gobo está parado!

Possível solução: O principal erro está na ausência do bloco “Quando clicar em bandeira verde” na programação do Gobo. Isso significa que não há um evento que dá início ao programa, que não vai fazer o Gobo dançar. Para resolver isso, uma alternativa simples é adicionar o bloco em questão no início da programação.

Desafio 2: *Scratch* está com Preguiça?

O programador deste algoritmo queria que o gato do *Scratch* virasse uma cambalhota ao pressionar a tecla “espaço”, mas ele simplesmente não se move.

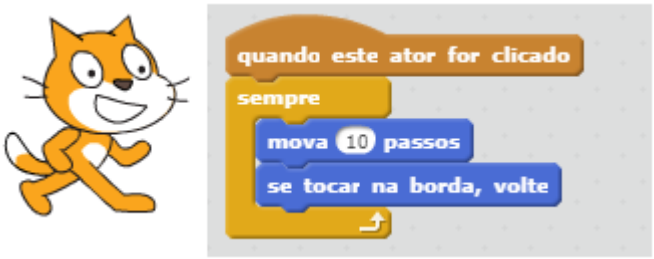
Scratch está com preguiça?

Possível solução: Cada equipe deve compreender que o programa está certo, o gato do *Scratch* está girando 360°, mas ele faz isso tão rápido que não conseguimos perceber. Isso pode ser consertado, por exemplo, adicionando intervalos de tempo entre cada “gire 90 graus”.

Desafio 3: De Cabeça para Baixo!

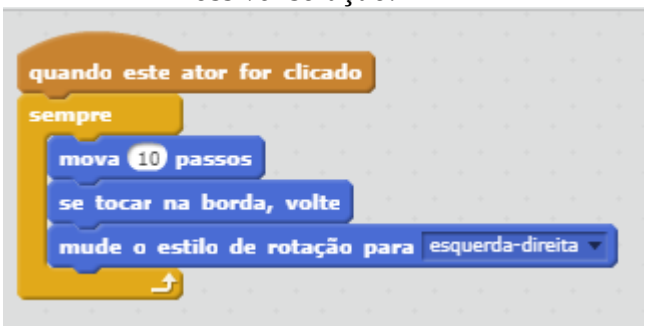
Neste projeto, o gato do *Scratch* deveria ir de um canto ao outro da tela. O problema é que ele faz isso e quando volta, está de cabeça para baixo!

De cabeça para baixo!



Possível solução: o principal problema deste programa pode ser resolvido com um bloco que caracteriza seu movimento.

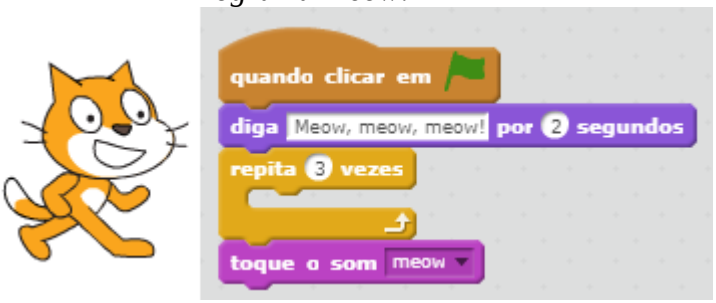
Possível solução.



Desafio 4: Meow!

A intenção do programa era que o gato do *Scratch* miasse de duas formas: com o balão de fala e através do som, que repetiria o miado 3 vezes. Algo no programa está fazendo com que ele reproduza o som somente depois do balão, e não ao mesmo tempo, e o som está sendo reproduzido somente uma vez.

Programa Meow!



Possível solução: o bloco “toque o som meow” deve estar dentro do comando “repita 3 vezes”. Além disso, deve ser trocado pelo bloco “toque o som meow até o fim”, para que o programa não execute 3 miados ao mesmo tempo. Para fazer com que o som seja executado ao mesmo tempo dos balões, existem algumas alternativas como dividir o programa em dois, ou trocar o bloco “Diga meow, meow, meow! por 2 segundos” por “Diga meow, meow, meow!”.

Encontro 4 – Jogo de Corrida!

A criação de jogos como uma ferramenta educacional e motivacional é o recurso a ser explorado neste encontro. A proposta é simples: criar um jogo de corrida onde existirão duas ou mais personagens, e, quando uma determinada tecla for pressionada, essa personagem irá se mover.

Descrição da Atividade

A turma inteira e o professor irão realizar a atividade em conjunto, com o auxílio do projetor, seguindo os preceitos do Pensamento Computacional.

É importante realizar uma lista de coisas que devem acontecer neste jogo de corrida e descrever como ele vai funcionar. Vamos aqui descrever cada parte do projeto, deixando aberto para qualquer alteração que possa vir a ser necessária. A criatividade conta muito!

Utilizando a habilidade de decomposição de problemas, vamos dividir nosso projeto em pequenas tarefas, para que depois elas formem a nossa solução.

Determinando os Competidores e seus Movimentos

As personagens do jogo de corrida são os competidores. Eles podem ser representados por personagens do próprio *Scratch* ou podem ser desenhados com a função de edição do programa.

No nosso caso, fizemos um círculo preto e um círculo verde, respectivamente objeto1 e objeto2.

Objetos 1 e 2.

Cada objeto deve se movimentar uma determinada distância a cada vez que uma tecla for pressionada. Isso pode ser feito facilmente com o comando “adicione 10 a x”.

Comando: adicione 10 a x.

Para cada competidor, será determinada uma tecla diferente do teclado. (Exemplo: espaço e seta para a direita).

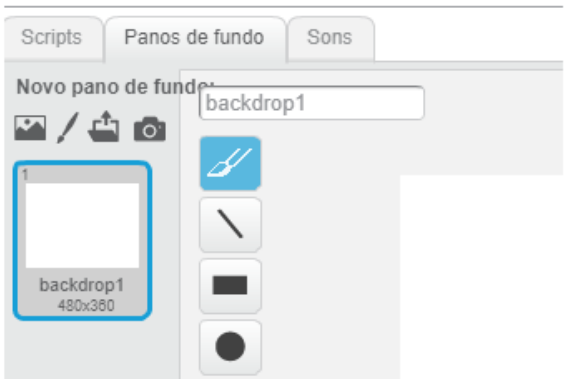
Pista de Corrida

Agora os competidores já "correm" pela tela. É necessário então, fazer a pista de corrida. Para isso, vamos acessar o palco, clicando ao lado dos atores, no local indicado (*Figura 28*).

Localização do palco.

Em seguida, selecionamos a aba "Planos de Fundo" (*Figura 29*) e pintamos nossa pista de corrida.

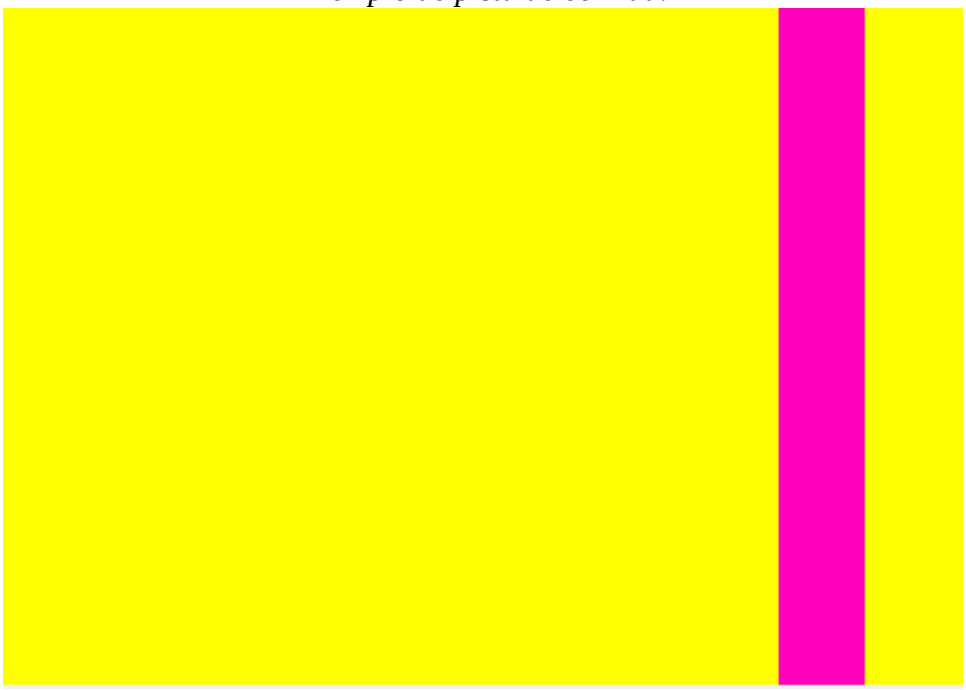
Aba de planos de fundo.



Devemos pensar como deve ser nossa pista de corrida. Discutimos com os estudantes alguns pontos importantes, como as cores, o formato e o tamanho, e, mais especificamente, como o programa vai detectar qual competidor ganhou a corrida.

Dessa maneira, nossa pista de corrida deve ter uma cor que determine a linha de chegada, pois é a partir da cor que o programa vai detectar qual objeto encostou primeiro na linha. Um exemplo simples de pista de corrida é um cenário de uma cor com um retângulo diferente.

Exemplo de pista de corrida.



Mecânica do Jogo

Definidos o cenário e os competidores, precisamos determinar como estes irão interagir um com o outro. Mais especificamente, precisamos encontrar uma maneira de determinar o vencedor.

Uma forma simples de fazer isso é utilizando o sensor de cor do *Scratch*, que determina quando o ator está encostando na cor especificada, neste caso, a cor magenta (*Figura 31*).

Utilizando o sensor de cor.

Agora a turma deve escolher como iremos identificar qual dos competidores ganhou. Algumas sugestões são: utilizar um som para cada competidor, utilizar os blocos de aparência, trocar o plano de fundo.

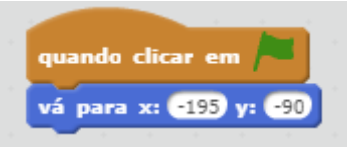
Customização

A turma criou o "jogo base" de corrida em conjunto, agora o desafio é que, separados em grupos de três componentes, os estudantes customizem seus jogos com outras funções. No final da aula, estes irão apresentar suas mudanças.

Algumas sugestões de desafios:

- Função de voltar os competidores para a posição inicial;

Exemplo de customização para voltar à posição inicial.



- Anúncio de que o jogador venceu através de um evento, que repercute nos scripts dos outros objetos;

Exemplo de customização com anúncio do vencedor.

- Adicionar a possibilidade de se movimentar para cima e para baixo;
- Pista de corrida com curva.

Apresentando o Jogo

Se a escola tiver disponibilidade e espaço, a fim de incentivar a divulgação dos resultados dos encontros dos estudantes, este jogo pode ser apresentado pelos criadores

para turmas em sala de aula ou até mesmo para a comunidade escolar (pais, professores) em eventos.

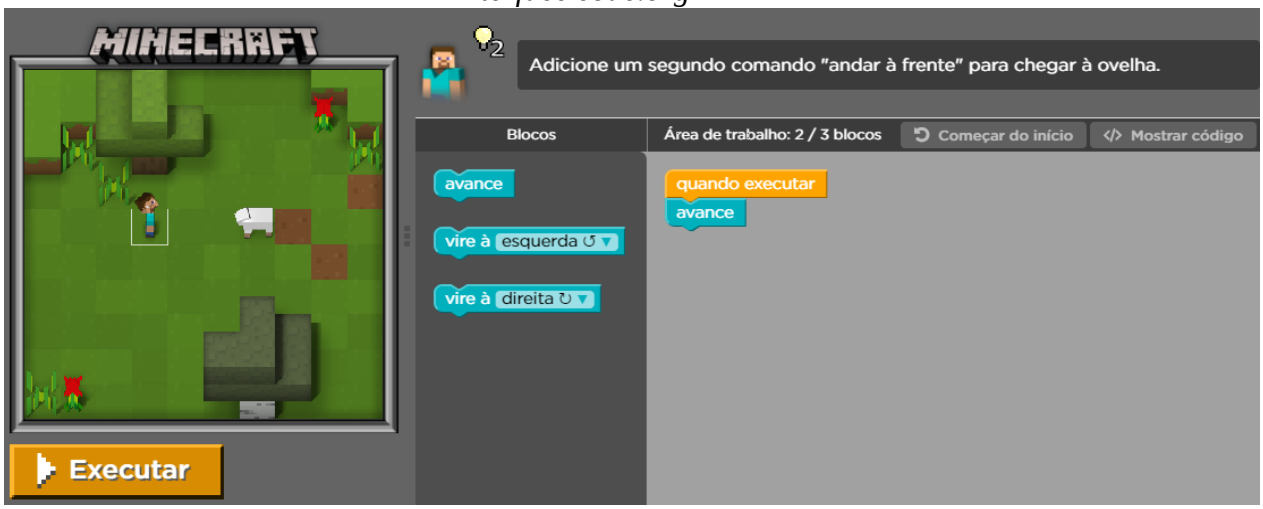
Encontro 5 - Hora do Código

A organização *code.org* é uma organização não lucrativa que se dedica a fazer a Ciência da Computação cada vez mais acessível nas escolas e ambientes de aprendizagem. Em seu *website*[3] encontramos diversas atividades, entre elas a Hora do Código.

A Hora do Código é um curso desenvolvido para que estudantes dediquem uma hora do seu dia para aprender a programação através de desafios simples que envolvem figuras de sua cultura (como, por exemplo, o jogo *Minecraf*t e as personagens da animação *Frozen*).

A proposta desta atividade é que os estudantes realizem a atividade "Aventureiro de *Minecraft*"[4]. Esta atividade é guiada pelo próprio jogo em sua interface, que lança os desafios e explica como funciona o ambiente de programação.

Interface code.org



O objetivo desta atividade é mostrar a diversidade de possibilidades com programação – dentre elas a programação de jogos muito populares. A inserção desta atividade na sequência didática surgiu da necessidade de mostrar outras formas de programar e para apresentar esta plataforma aos estudantes que demonstrarem interesse em explorar outros tipos de programação em casa.

[3] https://code.org/
[4] https://studio.code.org/s/mc/stage/1/puzzle/1

Encontro 6 – Debug it! 2.0

Com objetivos similares aos do terceiro encontro, serão propostos novos desafios para que os estudantes encontrem e depurem os erros de alguns programas

Descrição da Atividade

Separados em grupos, os estudantes irão receber programas com seus respectivos erros (*bugs*). Eles devem encontrar a parte do programa que está causando este erro e consertá-la.

Será solicitado que eles registrem em um pequeno formulário qual era o erro do programa e como ele foi corrigido.

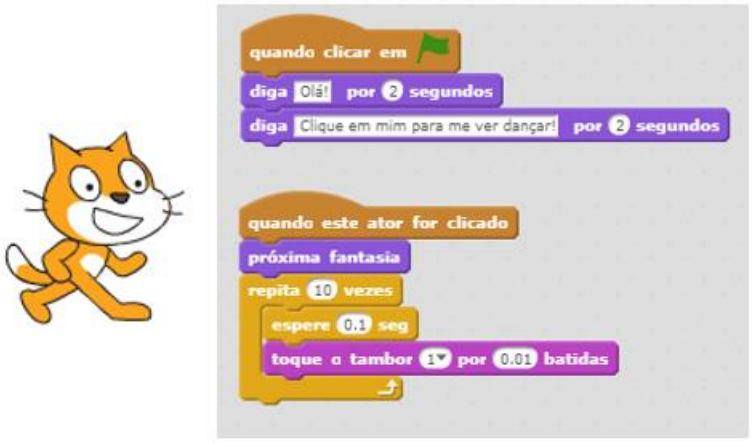
No final da aula, serão discutidas as soluções.

Alguns exemplos de programas com erros são apresentados a seguir:

Desafio 1: Dança?

Na programação, o gato do *Scratch* convida o usuário do programa para ver sua dança, mas ao clicar nele, ele apenas se move uma vez, e o som de tambores continua com ele parado. Como podemos consertar isso?

Dança?



Possível solução: Podemos simplesmente colocar o bloco “próxima fantasia”, que dá o movimento ao gato, dentro do ciclo “repita 10 vezes”.

Desafio 2: Pega-Pega

Na programação, Pico e Nano estão brincando de pega-pega. Quando Pico encosta em Nano, deveria dizer “Sua vez!”, mas isso não acontece. O que deu errado?

Pega-pega.

PICO

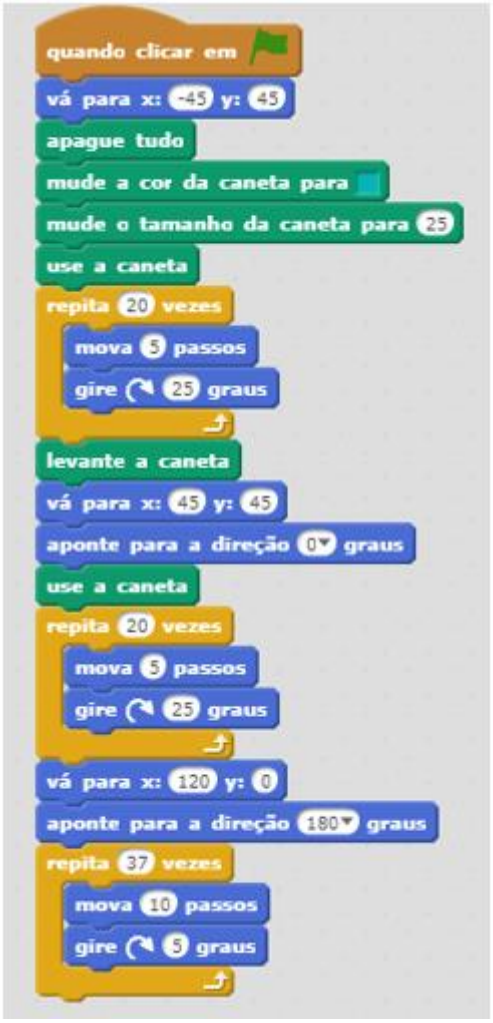
NANO

Possível solução: Pico estará “preso” no ciclo “repita até que tocando em borda”, e assim não executará a parte do programa “se tocando em Nano, então” até que seja tarde demais (depois de tocar na borda). Basta colocar o ciclo “se tocando em Nano, então diga Sua vez!” para dentro do ciclo “repita até que tocando em borda”.

Desafio 3: Rosto Feliz

Este programa deveria desenhar um rosto feliz, mas ele está conectando um dos olhos à boca. O que podemos fazer?

Rosto feliz.

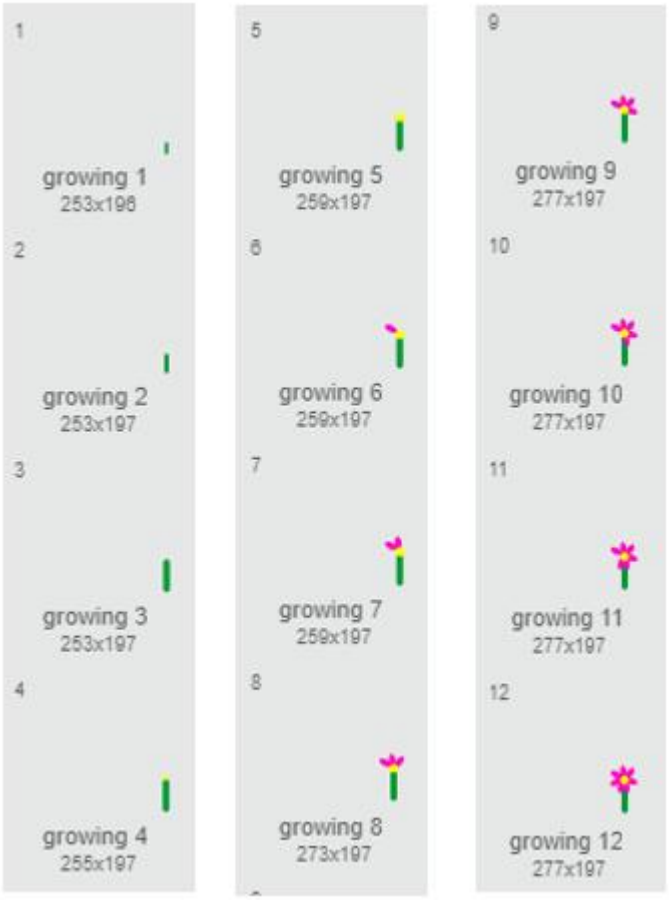


Possível solução: é preciso levantar a caneta ao ir da posição do olho até a posição onde a boca será desenhada. Para isso, inserimos blocos de “levante a caneta” e “use a caneta” nos locais adequados.

Desafio 4: Florescendo

Este programa deveria parar quando a flor está crescida, mas ele continua repetindo a animação sem fim. Como podemos solucionar este problema?

Florescendo.



Possível solução: A troca das fantasias da personagem será executada até que o número da fantasia seja maior do que 12. Como temos somente 12 fantasias, isso nunca irá acontecer. Para corrigir o problema, trocamos o número 12 por 11.

Podemos também trocar o bloco de operação “maior que” por “igual a”. Assim, quando o número da fantasia for exatamente 12, o programa irá parar.

Desafio 5: Feliz Aniversário!

Esta animação deveria tocar o tema “parabéns pra você” e, ao acabar a música, deveria aparecer a instrução de “Clique para apagar as velas!”, mas essa instrução aparece durante a música. Como podemos consertar este erro?

Feliz Aniversário!

Possível solução: trocar o bloco “toque o som birthday” por “toque o som birthday até o fim”.

Encontro 7 - Projeto Compartilhado

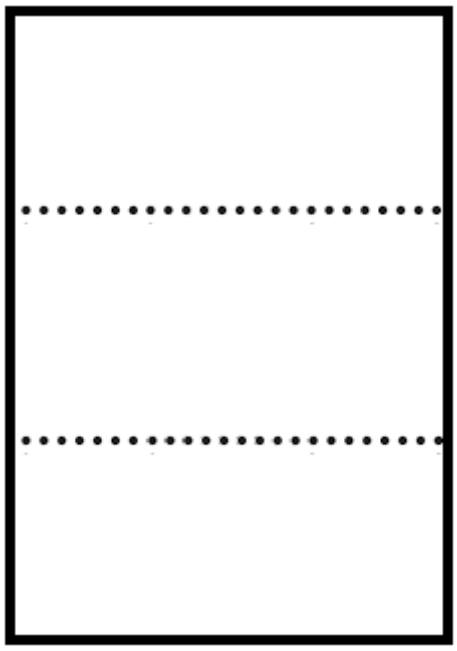
Nesta atividade, os estudantes irão criar projetos onde cada um vai adicionar um pouco do que conhece.

Descrição da Atividade

Para iniciar, os estudantes serão separados em duplas ou trios, dependendo do número de computadores. Cada grupo deve ficar com um computador para si. Será distribuído para cada equipe uma folha tamanho A4 e uma caneta hidrográfica colorida (cada caneta deve ter uma cor diferente).

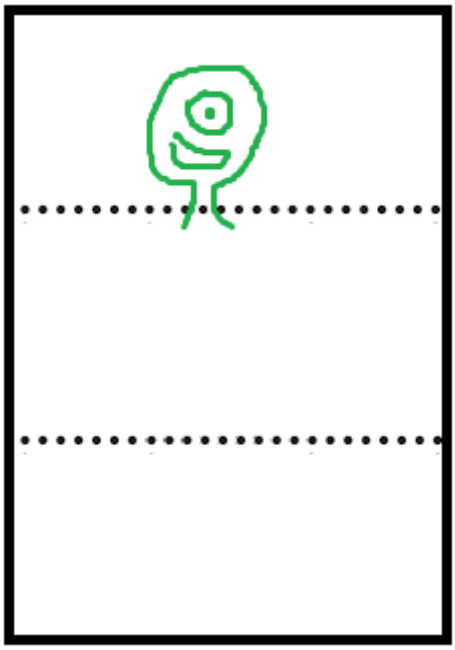
Os estudantes serão orientados a dobrar a folha em 3 partes.

Dobraduras da folha.



Nesta folha, devem usar sua imaginação e desenhar na primeira parte a cabeça de um monstro, de maneira que o pescoço fique na segunda parte, conforme exemplo.

Exemplo de desenho de monstro.



Em seguida, irão dobrar a folha, de maneira que seu desenho esteja escondido, e a turma vai fazer um rodízio de desenhos. O desenho deve ser entregue ao próximo grupo. O grupo não deve espiar o trabalho do anterior, mas deve agora fazer um corpo para este monstro a partir do pescoço que foi deixado.

De maneira semelhante, devem fazer até a cintura do monstro, e deixar as pernas para a próxima equipe finalizar.

Ao final da atividade, deve ser feita uma reflexão de como o trabalho em grupo pode ser divertido, e sobre os desafios de continuar e finalizar o trabalho de outra pessoa.

Agora os estudantes serão desafiados a fazer algo parecido, mas no *Scratch*! Os grupos irão iniciar uma animação no *Scratch*. Terão 10 minutos para iniciar o trabalho. Ao final destes 10 minutos, o professor irá anunciar que acabou o tempo e devem trocar de mesa, e continuar o trabalho do outro grupo. Repetirão o rodízio até que retornem ao seu computador (dependendo do tamanho da turma, pode ser necessário terminar a atividade antes).

No final, serão apresentados no projetor os projetos, e os grupos que iniciaram cada projeto devem descrever sua ideia original e o que acharam do resultado. Também serão estimulados a falar sobre a dificuldade de interpretar o projeto dos outros.

Encontro Final: Jogo do Labirinto

Será lançado um desafio que une os conceitos que foram trabalhados durante os encontros anteriores. Nesse sentido, ele serve como uma avaliação do desenvolvimento da turma em relação aos objetivos de aprendizagem.

Para este encontro, cada grupo deve criar, no *Scratch*, um jogo de labirinto: um jogo onde você controla uma personagem que deve solucionar o caminho de um labirinto sem encostar na parede.

A descrição desta atividade é a mais curta, pois é um deasfio para os estudantes criarem o jogo “do zero”. Será um grande teste de suas percepções e estratégias.

Descrição da Atividade

Separados em duplas ou trios, os estudantes receberão o desafio: criar um jogo de labirinto:

Uma personagem controlada pelo usuário deve percorrer um labirinto. Ao chegar no final, deve ser anunciado que o jogador venceu. Se o jogador encostar na parede, deve retornar ao início.

Os estudantes deverão criar os cenários e a personagem e utilizar as cores para diferenciar a parede do labirinto. Devem ser estimulados a fazer o máximo que conseguirem e a customizar o jogo conforme acharem melhor.

Durante esta atividade o professor poderá auxiliar, mas deve cuidar para que apareçam as aprendizagens e conceitos das atividades anteriores.