

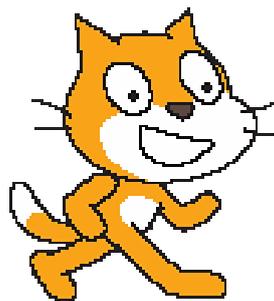


Jovem Engenheiro

ESCOLA DE NOVAS TECNOLOGIAS



KIDS MAKER



SCRATCH

e



Jovem Engenheiro Tecnologia Educacional Ltda
Rua Graça Aranha, 133
Bairro Fundação
São Caetano do Sul – SP
Cep 09520-240
PABX:- 55 11 42284466

Este material foi produzido no Brasil pela Jovem Engenheiro Tecnologia Educacional Ltda.

Esta publicação é parte integrante do Programa Jovem Engenheiro.

É expressamente proibida a utilização deste material sem o respectivo contrato entre a Jovem Engenheiro Tecnologia Educacional Ltda, e a instituição de ensino, e/ou empresa franqueada, salvo o caso de uso pelo aluno em sua residência.

A publicação, a cópia e a reprodução integral ou parcial, do conteúdo e/ou a utilização com finalidade outra, diferente daquela pactuada em contrato, somente será autorizada mediante permissão expressa e por escrito.

Este material foi concebido e elaborado e impresso de acordo com os novos BNCC.

O Software Scratch foi desenvolvido pelo MIT – Massachusetts Institute of Technology

www.jovemengenheiro.com.br - 2019

Novas Tecnologias

No futuro os cidadãos devem se dividir em duas categorias: os tecnologicamente passivos e os tecnologicamente ativos; o primeiro grupo utiliza a tecnologia, mas não detém o conhecimento estabelecendo assim o uso autômato da tecnologia, a segunda categoria será de cidadãos com capacidade de avaliar e transformar a informação em conhecimento, criticando, analisando e argumentando.

Ensinar e aprender implica em interação, disputa, aceitação, rejeição, caminhos diversos, percepção das diferenças, uma busca constante de todos os recursos tecnológicos disponíveis na ação de conhecer.

A aprendizagem segue um caminho que não é linear, mas uma trama de relações estabelecidas pelos diferentes atores que dela participam.

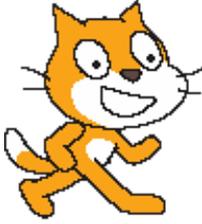
A busca pela integridade entre o discurso da aprendizagem e as ações que podem favorecê-la junto aos nossos alunos nos compelem ao investimento constante, físico e lógico, em nossos cursos e projetos de enriquecimento curricular.

Novas ferramentas estão sendo utilizadas como verdadeiros recursos para resolver com êxito diferentes tipos de problemas, nas mais variadas situações, onde a aprendizagem deve desenvolver-se em um processo de negociação de significados, novas formas de compreender e interpretar a realidade, questionar, discordar e propor novas soluções.

"O objetivo primordial é introduzir conhecimentos científicos e tecnológicos desde o início da educação escolar, mantendo a linha de progressão curricular até o ensino médio".

Dr. Seymour Papert – O pai da teoria construcionista

Pesquisador do Media Lab do MIT



Software Scratch

Curso de Games e Animação

Desenvolvido pelo laboratório do Massachusetts Institute of Technology (MIT), o software Scratch é muito mais acessível que outras linguagens de programação, por se utilizar de uma interface gráfica que permite programas sejam notados como blocos.

Por não exigir o conhecimento prévio de outras linguagens de programação, ele é ideal para pessoas que estão começando a programar e foi desenvolvido para ajudar no aprendizado de conceitos matemáticos e computacionais.

Com ele é possível criar histórias animadas, jogos e outros programas interativos.

Muitas vezes os games são associados apenas ao entretenimento, sem benefício educativo para os jogadores, Sendo assim, trazemos uma proposta de unir o desenvolvimento de games ao aprendizado de matemática.

Além de desenvolver a lógica de programação estudamos o conteúdo matemático do qual trata o jogo. Para isso, foi utilizado o Scratch que possui uma interface gráfica baseada em blocos de instruções onde o usuário vai agrupando-os de acordo com os seus objetivos.

Constatamos que o interesse pela disciplina de matemática aumentou consideravelmente além de despertar o interesse de criar jogos para outras disciplinas. Acreditamos que tal pesquisa traz contribuições significativas para relação entre games e educação.

Traduzido do site do Mit.

COMO OBTER O SCRATCH?

O download do **SCRATCH 2.0** está disponível em seu site oficial. Porém, antes de baixá-lo é necessário fazer o download do **ADOBE AIR** também disponível no mesmo site.

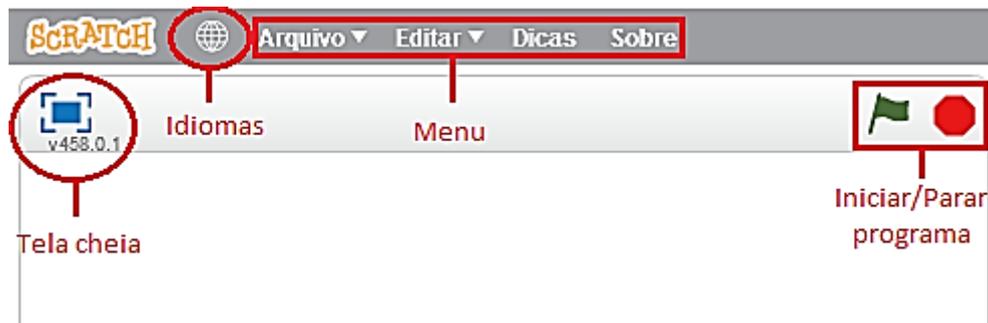
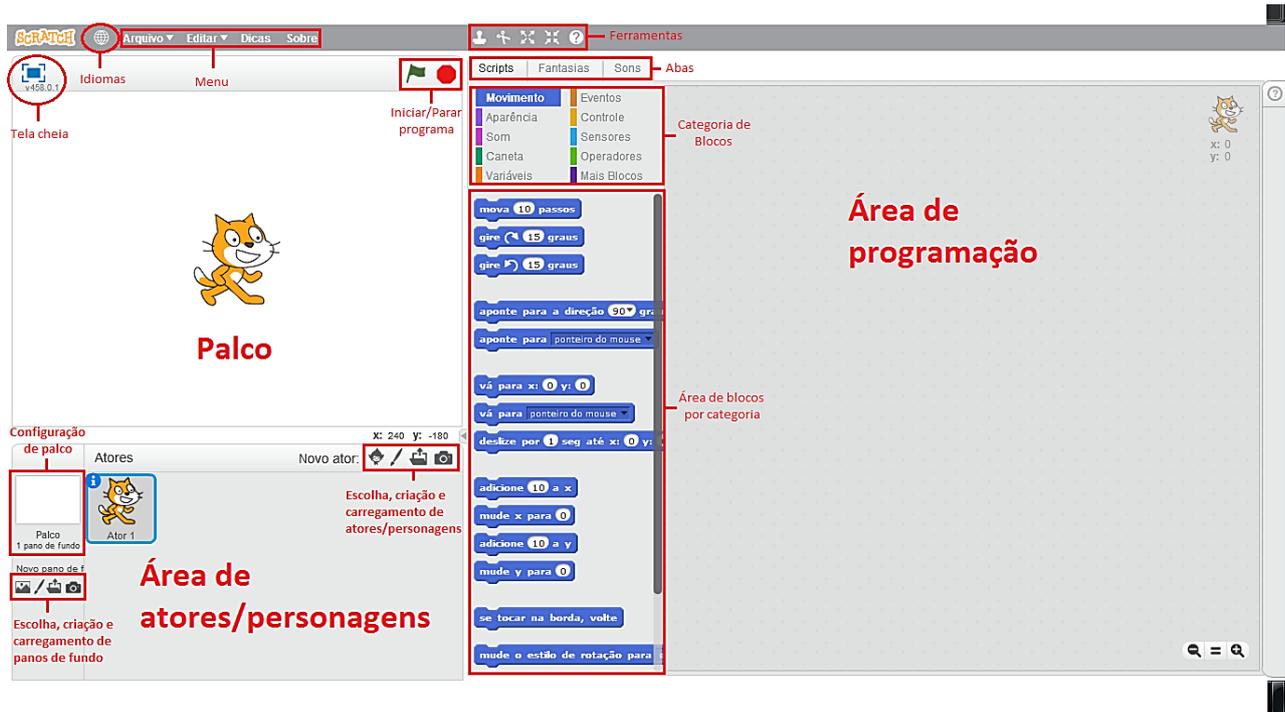
Basta acessar o site no link <https://scratch.mit.edu/download> e seguir os seguintes passos.

A tela Abaixo vai aparecer no site do MIT

	
Adobe AIR	Editor Desconectado do Scratch
Se ainda não a tem, descarregue e instale a última versão do Adobe Air	De seguida, descarregue e instale o Editor Desconectado do Scratch 2.0
Mac OS X - Descarregar	Mac OS X - Descarregar
Mac OS 10.5 e Mais Antigos - Descarregar	Mac OS 10.5 e Mais Antigos - Descarregar
Windows - Descarregar	Windows - Descarregar
Linux - Descarregar	Linux - Descarregar

1ª AULA – FAZENDO UM TOUR!

Explore cada parte do **SCRATCH 2.0**, praticando é que se aprende!



- **Idiomas**: Há vários idiomas disponíveis. Selecione o seu.
- **Menu**: No *menu* há 4 opções:
 - **Arquivo** onde fazemos novos projetos, salvamos os projetos finalizados e carregamos projetos já realizados anteriormente;
 - **Editar** onde há a opção *recuperar* o que torna possível reaver imagens, atores e blocos apagados acidentalmente;
 - **Dicas** onde ao clicar abrirá uma aba de jogos e animações para você se guiar com algumas programações pré-determinadas;
 - **Sobre** onde possibilita acesso ao site do software. Explore!
- **Palco**: É o local onde seu game e/ou sua animação será executado(a) após ser programado(a).

- **Tela cheia:** Possibilita visualizar o seu game ou animação de forma ampliada em sua execução.
- **Iniciar/parar programa:** São os ícones de **BANDEIRA VERDE** e **PLACA VERMELHA**. Após desenvolver todo o programa, você inicia ou encerra seu SCRIPT.



- **Escolha, criação e carregamento de pano de fundo:**

 Ao clicar neste ícone, abrirá uma página com vários panos de fundo de diferentes temas. Divirta-se com eles!

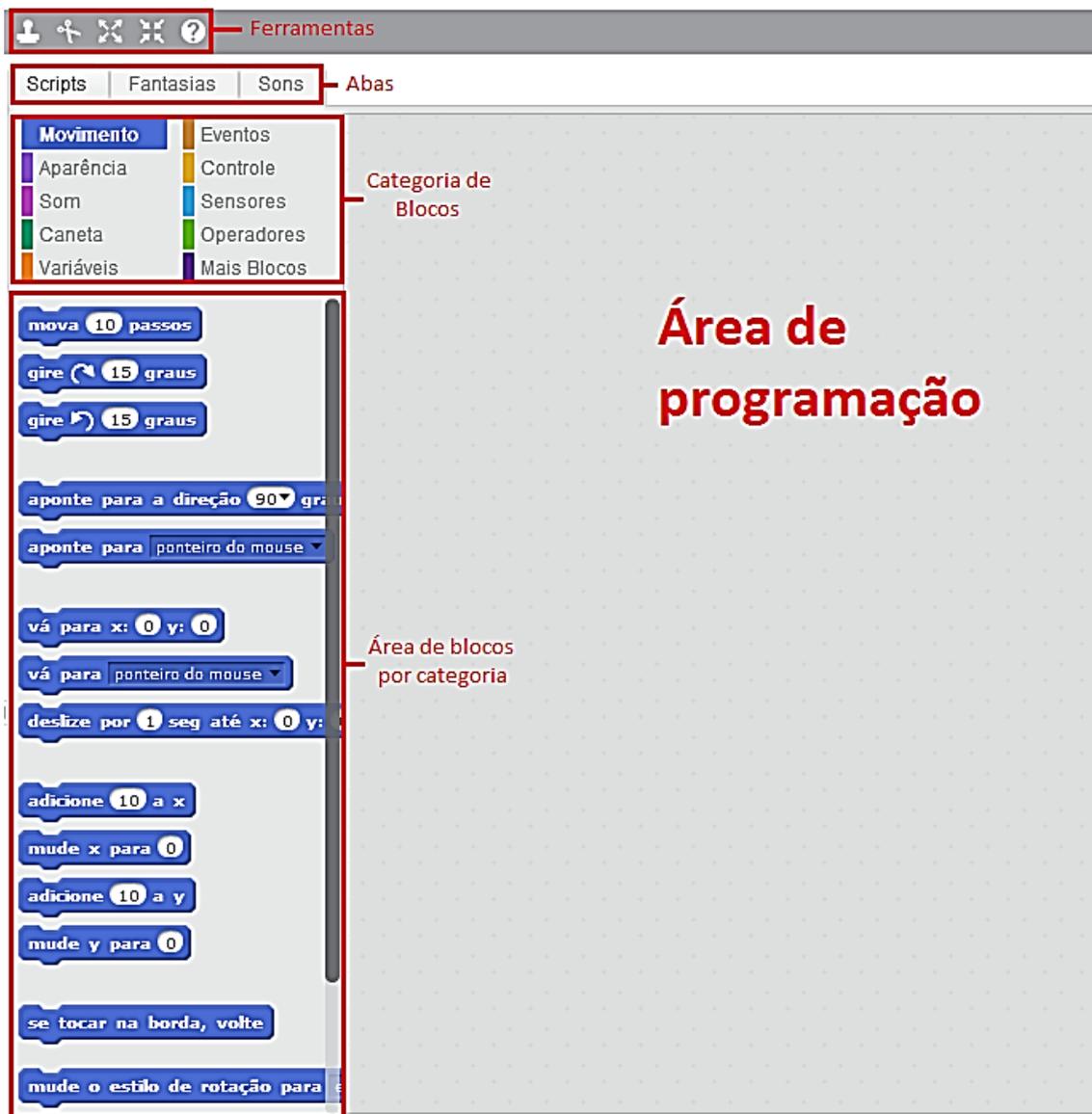
 Ao clicar neste ícone, você poderá criar seu próprio pano de fundo. Junte seus amigos e sejam criativos!

 Ao clicar neste ícone, você poderá *carregar* imagens. Basta escolher uma imagem na web, salvar ela em alguma pasta e através desse ícone você consegue carregá-la no **SCRATCH 2.0**.

 Ao clicar neste ícone é possível, através de uma *webcam*, tirar uma foto sua e/ou dos seus amigos e utilizá-la como pano de fundo no seu projeto.

- **Escolha, criação e carregamento de atores/personagens:**

 Apenas este ícone é diferente dos de pano de fundo, pois ele tem a função de *escolher, criar e carregar ATORES/PERSONAGENS*.



- Ferramentas:

 Esta é a ferramenta *duplicar*, ou seja, você pode duplicar um pano de fundo, uma fantasia ou um ator. É duplicado também sua programação.

 Esta ferramenta tem a função de *apagar* atores, panos de fundo, fantasias e blocos de programação.

 Estas ferramentas tem as funções *umentar* e *diminuir*, respectivamente o tamanho dos atores/personagens. Basta clicar na ferramenta e em seguida sobre eles.

 Esta ferramenta tem a função de *ajuda do bloco*. Ao clicar nele e sobre algum bloco, abrirá uma janela mostrando como executa esse bloco.

Observação: As ferramentas *duplicar* e *apagar* possui atalhos. Basta clicar com o botão *DIREITO* sobre um ator, pano de fundo, fantasia ou bloco que se deseja duplicar ou apagar.

- **Abas:** Cada ator/personagem possui **SCRIPTS, FANTASIAS** e **SONS**.

Você consegue escolher sons da biblioteca, gravar e carregar algum som salvo em seu computador como músicas, por exemplo.

- **Categoria de blocos:** Há 10 (dez) tipos de blocos separados por cores distintas para facilitar a visualização da programação durante nossas aulas.

- **Movimento:** somente relacionados à movimentação de personagens;
- **Aparência:** é o visual de seu projeto, ou seja, mudança de cor, brilho, texto a serem falados por um determinado ator e etc.
- **Som:** programação de sons em seu projeto.
- **Caneta:** Possibilita a construção de linhas e imagens de acordo com o trajeto de um ator/personagem.
- **Variáveis:** Possibilita a “criação” de blocos, onde armazenam valores ou qualquer outra coisa.
- **Eventos:** Início de um script e comunicação com outros atores.
- **Controle:** Controla o tempo, quantidade, repetições e etc.
- **Sensores:** Orientação por cores, valores, ponteiro de mouse e outros.
- **Operadores:** Operações aritméticas, aleatoriedade e etc.
- **Mais blocos:** Permite criar blocos e determinar qual a função de cada um.

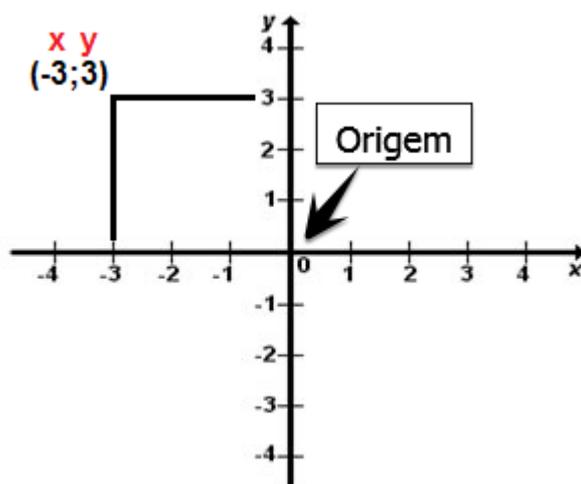
- **Configuração de palco:** Tanto os atores/personagens como o palco podem ser programados, ou seja, há scripts para panos de fundo e etc.

IMPORTANTE: Para colocar os blocos na área de programação, basta arrastá-los com o mouse e encaixá-los uns sobre os outros.

2ª AULA – PLANO CARTESIANO NO SCRATCH 2.0

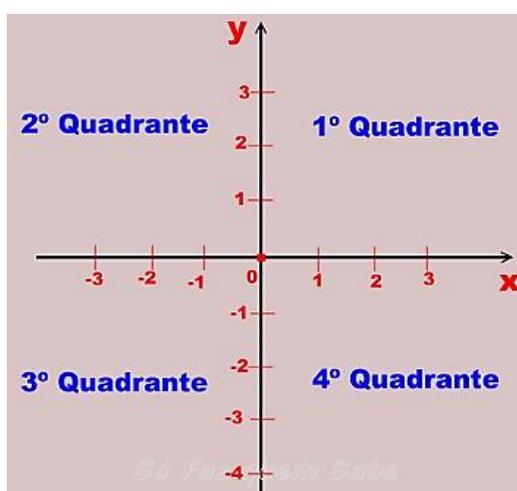
O que é o Plano Cartesiano?

O Sistema de **Coordenadas Cartesianas**, mais conhecido como **Plano Cartesiano**, foi criado pelo matemático René Descartes com o objetivo de localizar pontos. Ele é formado por dois eixos perpendiculares: um horizontal e outro vertical que se cruzam na origem das coordenadas. O eixo horizontal é chamado de abscissa (x) e o vertical de ordenada (y). Os eixos são enumerados compreendendo o conjunto dos números reais. Observe a seguir uma figura representativa do plano cartesiano:



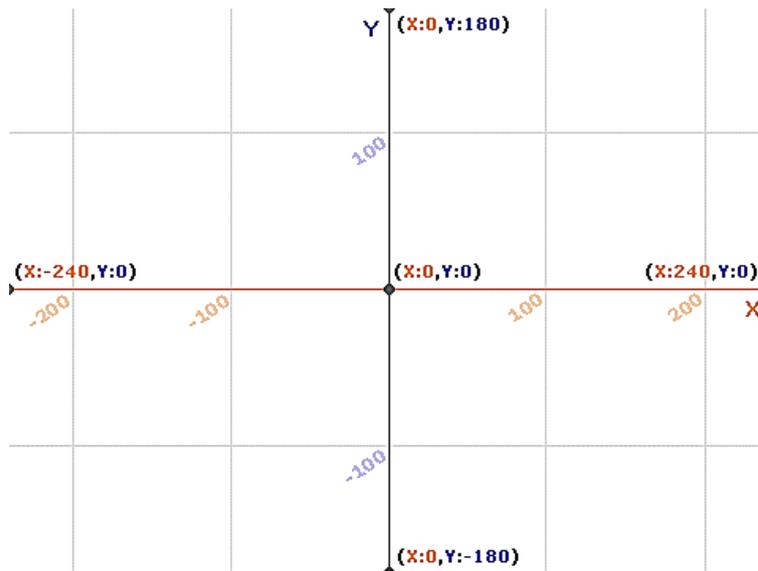
As coordenadas cartesianas são representadas pelos pares ordenados $(x;y)$. Em razão dessa ordem, devemos localizar o ponto observando primeiramente o eixo **x** e posteriormente o eixo **y**. Observando o par ordenado $(-3;3)$, podemos afirmar que $x = -3$ e $y = 3$.

Cada uma das 4 partes do Plano Cartesiano recebe o nome de **QUADRANTE** e são enumeradas em sentido anti-horário como mostra a figura a seguir:

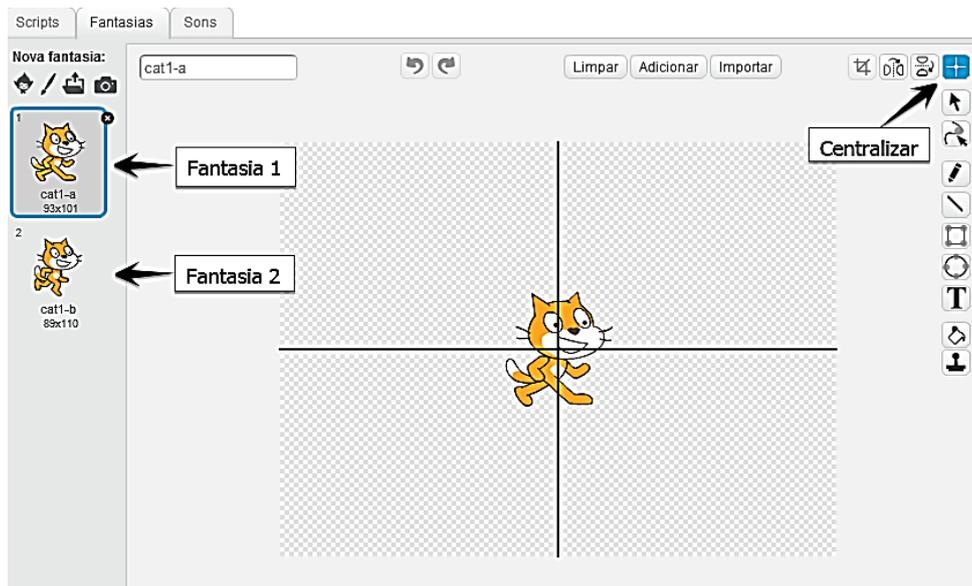


O plano cartesiano no SCRATCH 2.0.

A figura a seguir é um dos panos de fundo e representa o **Plano Cartesiano** definido pelo MIT:



Observe que no eixo **X** há um limite de pontos de **240 para a direita** e **-240 para a esquerda**. No eixo **Y** o limite é de **180 para cima** e **-180 para baixo**. Esses limites são exatamente o tamanho do palco. Você consegue determinar o centro de cada fantasia dos atores como mostra a seguinte figura:



Caso você determine que a **cabeça** do gato será o seu ponto central da fantasia e você programar o ator para ficar no centro do palco, a **cabeça** é quem ficará centralizada e não mais o seu corpo como na figura anterior.

Escolha o plano de fundo **xy-grid** localizado na categoria **outros**.

Antes de iniciarmos o projeto, verifique se o gato está contornado por uma faixa

azul



Caso não esteja, basta clicar no ator.

Programo-o para que ele fique no centro do palco. Utilize os blocos da seguinte forma:



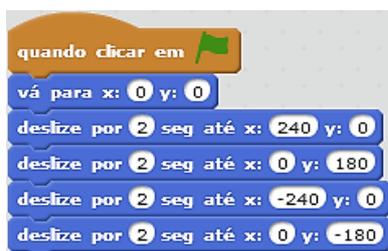
Ao clicar na **bandeira** observe que o gato está exatamente no centro do palco. Agora troque as coordenadas para (240;0), (0;180), (-240;0) e (0;-180). Em cada troca, clique na **bandeira** e observe as novas posições do gato no palco.

Observação: Lembre-se de que é você quem determina o ponto central do ator.

Após ter feito todas as trocas, volte as coordenadas para (0;0) e acrescente no programa o seguinte bloco utilizando 2 segundos e coordenadas (240;0):



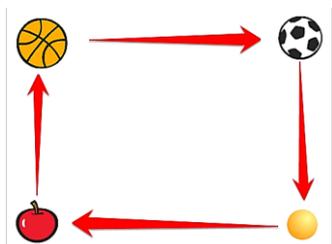
Duplicate os blocos e preencha-os com todas as coordenadas, conforme a figura:



3ª AULA - PRATICANDO

Você viu na aula anterior que o gato deslizou pelos quatro quadrantes em sentido anti-horário, correto? Agora escolha outro ator de sua preferência e faça com que ele deslize pelos quatro quadrantes, mas em sentido **horário**.

Agora adicione mais 2 atores (4 atores ao todo) de sua preferência, faça com que cada um deles deslize em sentido horário podendo se encontrarem **somente** no momento de partida, ou seja, no centro do palco, como ilustrado nas imagens abaixo:



DESAFIO

Faça com que os atores, ao clicar na **bandeira**, não parem de girar a menos que você clique na **placa vermelha**.

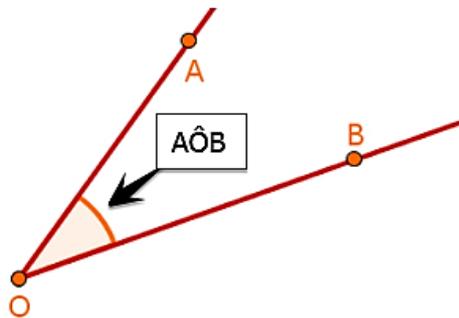
4ª AULA – DIRECIONANDO UM ATOR

Por que devo **direcionar** meu ator?

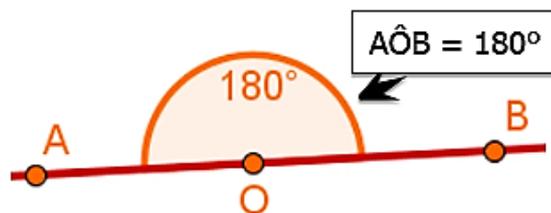
Nos exercícios anteriores você percebeu que os atores giravam em sentido horário e anti-horário, porém eles **não** mudavam a **direção** para a qual estavam virados. Imagine-se andando o tempo todo de costas! Por isso a importância do direcionamento dos atores, mas antes de você aprender essa nova etapa, deve-se ter em mente o conceito de **ângulos** e a utilização deles no **SCRATCH 2.0**.

O que é um ângulo?

Um **ângulo** é o conjunto de pontos formados por duas **semirretas** (lados do ângulo) que possuem o mesmo ponto de partida (**vértice do ângulo**). Veja a figura a seguir:



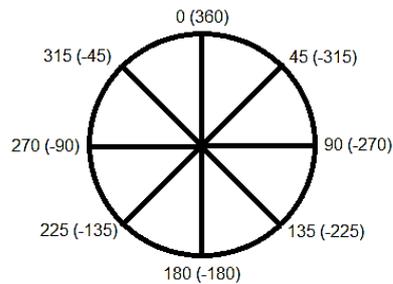
Também pode ser um número usado para medir a **abertura** dessas semirretas. Sendo assim, quanto maior o valor numérico atribuído a um **ângulo**, maior será a abertura entre as duas semirretas relacionadas a ele, como mostra a seguinte figura:



Os ângulos no SCRATCH 2.0

O que determina a direção dos atores no software são os ângulos.





No software **0** significa **para cima**, **90** significa **para a direita**, **180** significa **para baixo** e **-90** significa **para a esquerda**. Basta clicar na seta para baixo e selecionar uma dessas opções no menu suspenso. Para outras direções, digite o valor que você quiser na caixa branca de edição. Você pode até mesmo utilizar valores negativos! (Por exemplo, se você digitar **45** ou **-315**, ambos fará com que o ator aponte igualmente para nordeste.)

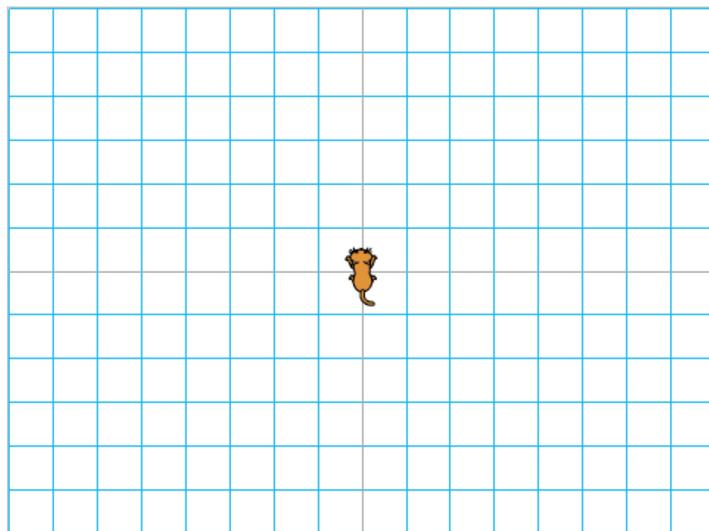
Observação: A direção corrente do ator pode ser encontrada na **área de informações do ator** ⓘ ou você também pode clicar na caixa de seleção ao lado do bloco **direção** localizado na categoria **Movimento**.

Agora você irá praticar esses novos conceitos que são bem simples, mas muito importantes. Escolha o pano de fundo **xy-grid-30px** localizado na categoria **outros** e escolha o ator **cat2** na categoria **animais**. Cada quadradinho deste plano de fundo possui 30 pontos em cada lado no plano cartesiano, ou seja, para fazer com que o gato ande pelas linhas deve-se **adicionar** ou **diminuir** 30 tanto na coordenada **x** (para direita/esquerda) quanto na **y** (para cima/baixo).

```

quando clicar em [bandeira]
  apague tudo
  levante a caneta
  vá para x: 0 y: 0
  aponte para a direção 0 graus
  deslize por 1 seg até x: 0 y: 30
  use a caneta
  aponte para a direção 90 graus
  deslize por 1 seg até x: 30 y: 30
  aponte para a direção 180 graus
  deslize por 1 seg até x: 30 y: -30
  aponte para a direção -90 graus
  deslize por 1 seg até x: -30 y: -30
  aponte para a direção 0 graus
  deslize por 1 seg até x: -30 y: 30
  aponte para a direção 90 graus
  deslize por 1 seg até x: 0 y: 30
  
```

Diminua o tamanho do gato e programe-o da seguinte maneira:



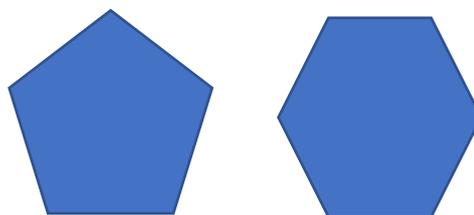
Conseguiu perceber qual a função da **caneta**? Ela foi utilizada para fazer o contorno de um quadrado **simétrico** (se dividir esse quadrado ao meio, ambos os lados devem ser espelhados). Agora troque os blocos **deslize** e **aponte** por **mova** e **gire**, respectivamente. Após fazer o gato percorrer o mesmo percurso, faça uma nova troca. Em vez de utilizar o **mova**, utilize **adicione a x** e **adicione a y**. O gato deve fazer o mesmo percurso em todas as trocas.

5ª AULA - PRATICANDO

Na aula anterior você aprendeu a programar um ator para desenhar um quadrado. Na aula de hoje você vai praticar os conceitos que você aprendeu e vai fazer o gato desenhar um retângulo e um triângulo, um ao lado do outro, ambos simétricos, isto é, se cortarmos a figura geométrica ao meio, ambos os lados devem ser iguais.

DESAFIO

Acrescente mais um personagem e faça o gato desenhar um pentágono (figura plana de 5 lados) e o outro desenhar um hexágono (figura plana de 6 lados). Os personagens devem desenhar ao mesmo tempo as figuras.



6ª AULA – MINHA PRIMEIRA ANIMAÇÃO

Chegou o momento de você aprender como fazer um ator **falar**, **pensar**, **aparecer** e **desaparecer**.

Na categoria **APARÊNCIA** há 2 blocos com a palavra **diga**. Com eles, você consegue fazer um ator dizer algo. Escolha um ator de sua preferência, programe-o da seguinte maneira e clique na **bandeira**.



Observe que a palavra **OLÁ** apareceu em um balão acima do ator no palco. Porém, este bloco faz a palavra ou frase ficar permanente no palco. O bloco



é muito mais utilizado durante as aulas, pois você é quem determina o tempo em segundos de permanência do texto no palco. Substitua-os e veja a diferença. Você pode utilizar quantos blocos quiser. Basta escrever e determinar o tempo de cada bloco.

Os blocos  e  funcionam da mesma maneira, sendo a diferença apenas no desenho do balão. Faça o teste!

Para fazer um ator aparecer e desaparecer utiliza-se  e , respectivamente. Acrescente à programação de seu ator o bloco **esconda** e clique na **bandeira**. Sim, o seu ator desapareceu!

IMPORTANTE: *Esconder* não significa *apagar*. O seu ator ainda existe, apenas está **escondido** no palco. Se você clicar mais uma vez na **bandeira** nada acontecerá e seu ator não aparecerá.

*Por que meu ator não aparece quando eu clico na **bandeira**?*

Porque na programação há apenas o bloco **esconder** e ele atendeu aos seus comandos. Para fazer com que ele apareça, acrescente o **mostre** antes do **diga** e clique na **bandeira** novamente. O seu ator *aparecerá*, *dirá OLÁ* e *desaparecerá* conforme está na programação.

7ª AULA - PRATICANDO

Na aula anterior, você aprendeu a fazer animações, onde o personagem fala, pensa e etc. Hoje você vai fazer uma animação utilizando 2 atores que se apresentam **juntos** dizendo: nome, idade e o que mais gosta de fazer. (Ex. Ator 1: *Meu nome é J.E., eu tenho 20 anos de idade e gosto de fazer parte do aprendizado das crianças!*).

Em seguida, acrescente mais um personagem totalizando 3 atores onde, ao clicar na **bandeira**, fiquem escondidos e ao aparecerem juntos apresentam-se dizendo: nome, idade e qual matéria da escola mais gosta.

DESAFIO

Faça um diálogo entre 4 atores onde todos iniciam escondidos e ao aparecerem, um a um, apresentam-se e conversem entre si. As falas devem ser ditas uma de cada vez.

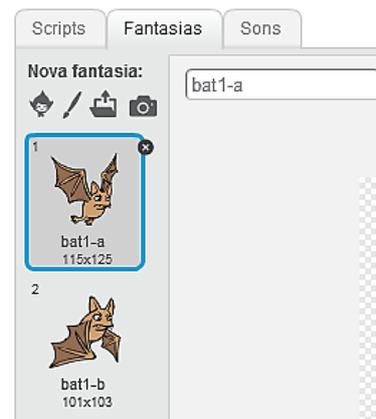
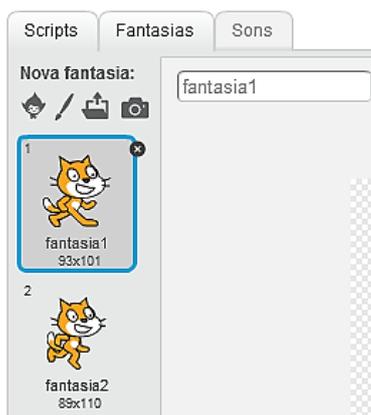
8ª AULA – DE NORTE À SUL

Hoje você aprenderá 2 novos conceitos muito importantes: a troca de **fantasias** e **panos de fundo**. A **fantasia** é responsável por dar a **impressão** de caminhada no gato ou batimento de asas do morcego, por exemplo.

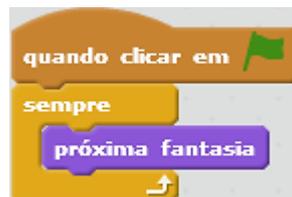
Como assim?

Para fazer o gato caminhar, ou seja, “mexer” as pernas e o morcego bater as asas é necessário uma troca de fantasias:

- Para o gato: uma fantasia com a perna **direita na frente** e a outra com as pernas **flexionadas**;
- Para o morcego: uma fantasia com as asas **levantadas** e outra com asas **abaixadas** como mostra as seguintes figuras:



Nesta aula você irá utilizar o gato e o morcego da imagem para praticar. Programe o gato e o morcego da seguinte maneira:



Clique na **bandeira** e observe os dois atores. Veja que eles estão trocando de fantasias **sem parar** e isso acontece porque o bloco **próxima fantasia** está dentro do **sempre**. Porém essa troca está sendo feita muito rapidamente e isso se resolve com o controle de tempo entre as

trocadas, utilizando . Ao acrescentar esse bloco dentro do **sempre**, essa troca de fantasias passará ser bem lenta e para ajustar a velocidade troque o **1** por **0.5 seg**. Escolha outros atores e divirta-se com as trocas de fantasias!

IMPORTANTE: O **SCRATCH 2.0 não** aceita vírgulas em números decimais, por isso o uso do **ponto**.

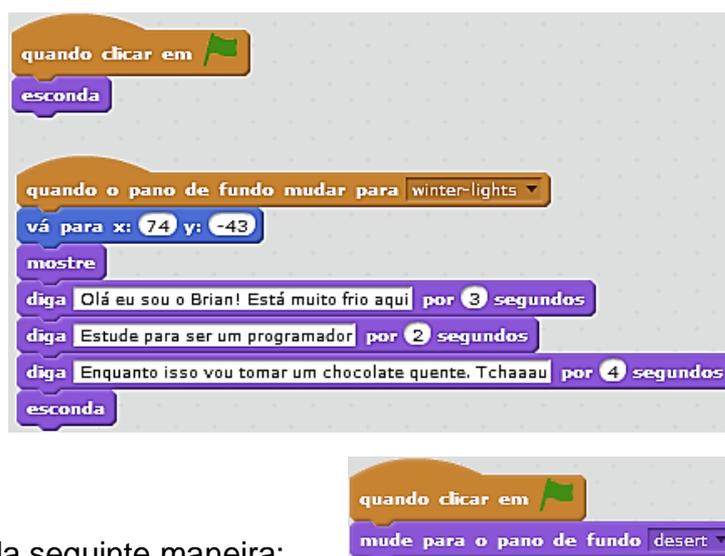
Pare sua animação e abra um novo arquivo, pois agora você aprenderá sobre as trocas de **panos de fundo**.

Escolha 2 panos de fundo: **DESERT** e **WINTER-LIGHTS**. Apague o gato e escolha dois atores: **HORSE 1** e **BEAR 1**.

Programme o **HORSE 1** da seguinte maneira:



Programme o **BEAR 1** da seguinte maneira:



Programme o **PALCO** da seguinte maneira:

Clique na **bandeira** e divirta-se com sua animação!

9ª AULA - PRATICANDO

Faça uma animação utilizando 2 atores e 3 panos de fundo distintos. A cada troca de pano de fundo os atores devem aparecer e esconder mantendo um diálogo coerente entre eles como na aula anterior.

Dica para sua animação: Faça uma animação simulando um diálogo entre você e seus responsáveis no caminho até a escola. Não se esqueça de utilizar atores e panos de fundo coerentes com a história.

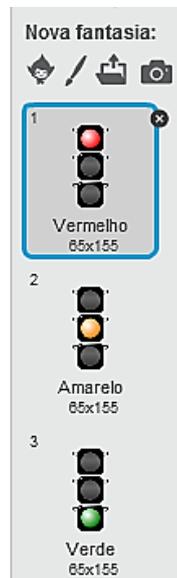
DESAFIO

Crie um pano de fundo simulando sua casa e crie outro simulando sua escola. Escolha um ator coerente e faça ele caminhar pelo palco como se estivesse indo em direção à escola. Faça as respectivas trocas de panos de fundo e fantasias do ator. O ator deve se apresentar e dizer onde está indo e se ele gosta das aulas de robótica ou não.

10ª AULA – BRINCANDO COM SCRATCH 2.0

Nessa aula você irá por em prática todos os conceitos aprendidos até hoje. O mundo inteiro sofre com problemas no trânsito e você está encarregado de reeducar esses motoristas infratores.

Escolha o pano de fundo **night city with street** e o ator **convertible 1**. Importe da pasta “**Material 10ª Aula**” apenas 1 dos faróis como **ator**. Os outros 2 importe como **fantasias do ator**. Sua área de fantasias deverá ficar dessa maneira:

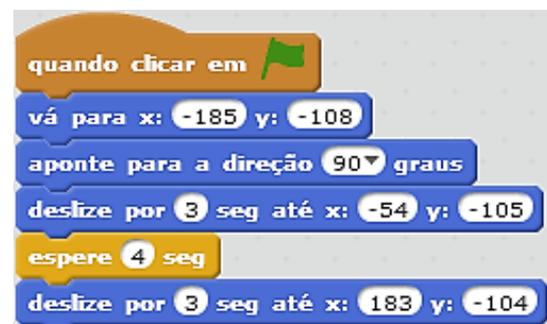


Observação: A ordem das fantasias **NÃO IMPORTA!** Caso seja necessário diminua o tamanho do farol no palco.

- Programe o farol da seguinte maneira:



- E o carro dessa maneira:



IMPORTANTE: Os valores das coordenadas podem ser alterados de acordo com sua preferência, ou seja, basta posicionar, com o mouse, o ator em qualquer lugar no palco e o próprio software encontra a coordenada exata daquele ponto do palco.

Clique na **bandeira** e veja como ficou sua animação.

11ª AULA – PRATICANDO

Na aula anterior, você fez um projeto onde aprendeu a programar um carro para respeitar as regras de trânsito. Hoje você vai dar continuidade nele então abra o projeto salvo e faça com que o carro ande pela pista, pare no farol, ande até o final da pista e volte fazendo o percurso inverso.

Faça também com que o carro volte ao chegar no final da pista, porém na faixa de cima.

DESAFIO

Faça uma animação com dois carros, partindo um de cada lado, parando juntos no farol e ao voltar troquem de faixa.

12ª AULA – MEU PRIMEIRO GAME

Antes de você iniciar o seu primeiro game, você deve ter em mente alguns conceitos importantes como **aleatoriedade** e “**Se Então Senão**”.

- **Aleatoriedade**: qualidade ou característica do que é aleatório; indeterminação, incerteza, casualidade. Ou seja, tudo o que você determinar como aleatório será uma incerteza, como, por exemplo, há essa opção na sua pasta de músicas ou aplicativo no seu celular. Você nunca saberá qual será a próxima música. A aleatoriedade pode ser aplicada a intervalo de tempo, espaço, quantidades e etc.



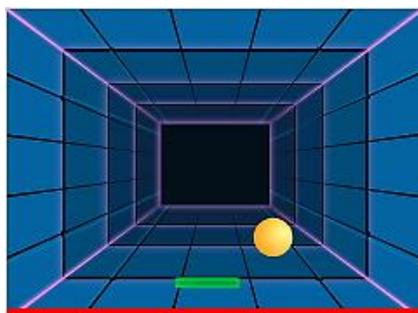
- **Se Então Senão**: O **se então** e o **se então senão** são blocos usados **somente** quando existe uma **condição**. Observe a situação a seguir para ficar mais claro o uso desses blocos:

SE eu tenho dinheiro **ENTÃO**
vou para a Disney
SENÃO fico em casa

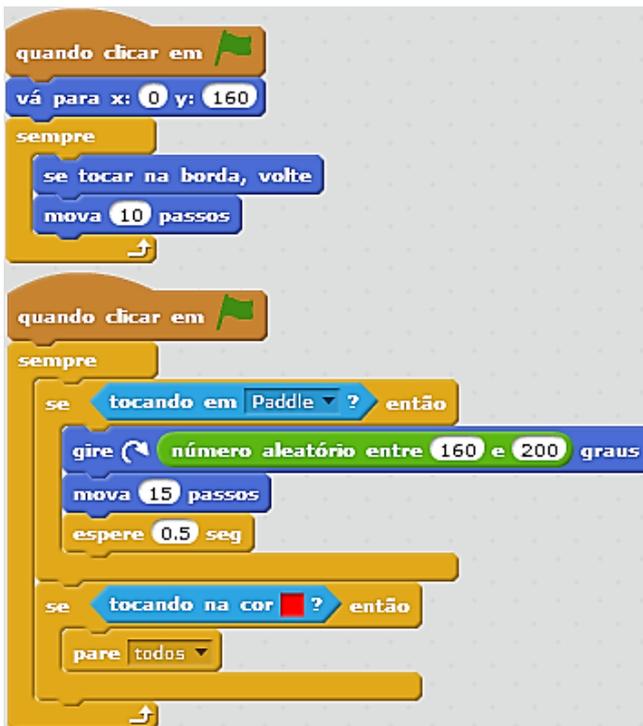
Agora você fará um game simples para entender esses novos conceitos na prática.

Escolha o pano de fundo **NEON TUNNEL** e os atores **BALL** e **PADDLE**.

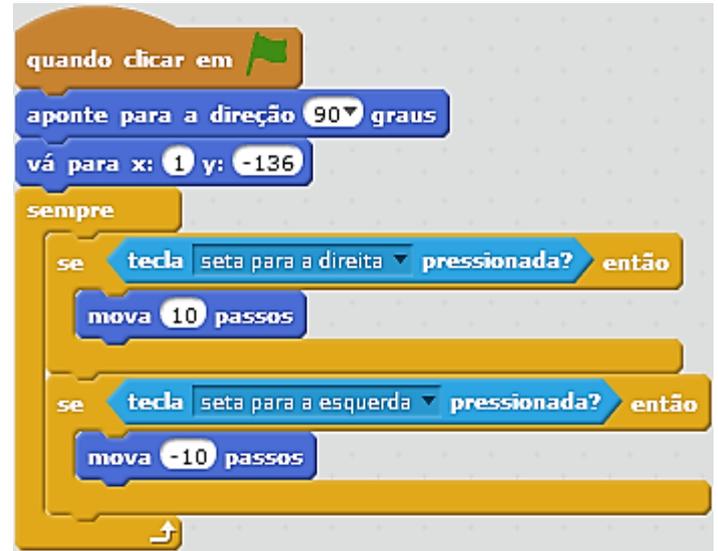
Antes de iniciar a programação, desenhe uma faixa **vermelha** na parte inferior do pano de fundo como na figura a seguir:



- Programe a bolinha da seguinte maneira:



- Programe o Paddle dessa maneira:



IMPORTANTE: Você poderá encontrar dificuldade em definir a cor do bloco **tocando na cor**. Basta clicar no quadradinho de cores do bloco e em seguida clicar sobre a cor na qual você deseja. No caso, é a barra **vermelha**.

Por que foi determinado -10 para o número de passos para a esquerda?

Por que o palco do software, como foi dito no início do curso, é um plano cartesiano e se você voltar na primeira aula verá que no sentido esquerdo do eixo x há somente números negativos.

Clique na **bandeira** e jogue o seu primeiro game!

13ª AULA - PRATICANDO

Na aula de hoje você vai dar continuidade no game da aula anterior. Abra o game salvo em seu computador, acrescente mais 2 bolinhas e elas devem trocar de cor a todo momento.

Em seguida faça com que cada vez que as bolinhas tocarem a cor vermelha, troque de pano de fundo e neste deve estar escrito **GAME OVER** ou algo do tipo.

DESAFIO

Acrescente panos de fundo, atores, fantasias e melhore o máximo que você puder o seu projeto. Este momento é livre para você.

14ª AULA – JOGO DAS MAÇÃS

Neste jogo, você também irá utilizar aleatoriedade, mas em local diferente. Você também irá utilizar **variáveis**, então antes de iniciar seu projeto entenda um pouco sobre esse conceito. Você irá estudar mais a fundo as variáveis nas próximas aulas.

Definição: (variável) inconstante; flexivo; que varia ou pode variar; elemento de um programa que armazena valores; elemento de uma função que pode assumir diferentes valores.

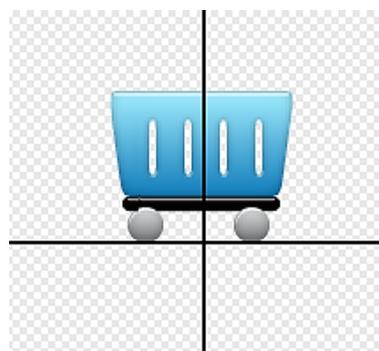
Como assim?

Quando você cria uma variável no **SCRATCH 2.0** é como se você estivesse criando uma gaveta que guardará valores ou qualquer outra coisa. O **placar** de jogo de futebol é um exemplo claro de variável, ou seja, a cada gol feito por um time é **acrescentado** 1 no placar. Esse placar é uma variável.

Escolha o ator **APPLE**, importe o **CARRINHO** como ator e o **PANO DE FUNDO** contidos na pasta **“Material 14ª Aula”**.

Clique na categoria **variáveis** e crie uma variável chamada **PLACAR**. Observe que apareceu o componente **Placar 0** no palco.

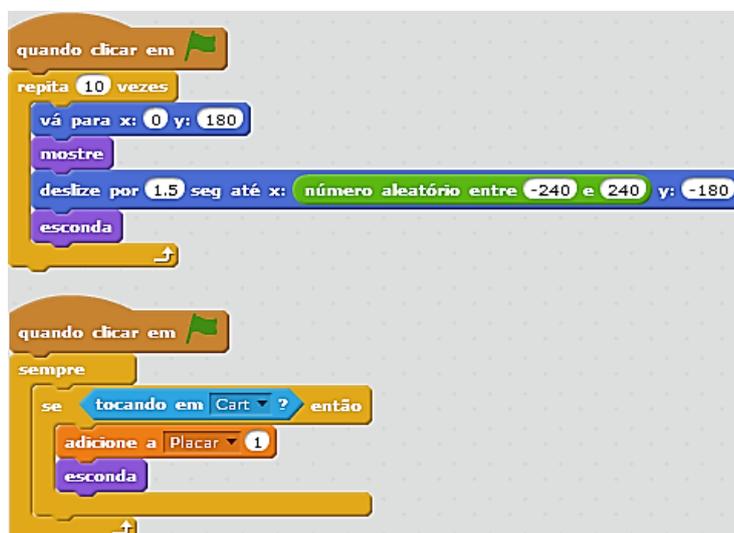
Defina o **centro** do carrinho como mostra a figura a seguir:



- Programe o carrinho da seguinte maneira:



- Programe a maçã dessa maneira:



Clique na **bandeira** e teste seu game.

15ª AULA - PRATICANDO

Na aula anterior você jogou bastante o game PEGA MAÇÃS e, provavelmente, você percebeu que, ao clicar na **bandeira** e tentou jogar 2 ou mais vezes, o placar **não zerou**. Abra-o novamente e tente com os conhecimentos que você adquiriu até agora a resolver este problema.

Em seguida, aumente o tempo de jogo, a velocidade de caída das maçãs e a velocidade do carrinho.

DESAFIO

Faça com que, ao atingir 30 pontos no placar, por exemplo, mude de pano de fundo e neste deve estar escrito **“VOCÊ VENCEU”**.

16ª AULA – TROCANDO IDEIAS

Você já aprendeu a fazer um diálogo entre atores, porém, quando há uma animação muito extensa, controlar o tempo de espera como foi feito na 5ª aula torna-se algo muito difícil e demorado. Existem formas mais fáceis de interação entre atores e é nessa aula que você irá aprender esse novo tipo de programação. Esse método também é usado para controlar os momentos de **“mostrar e esconder”** durante os games e animações.

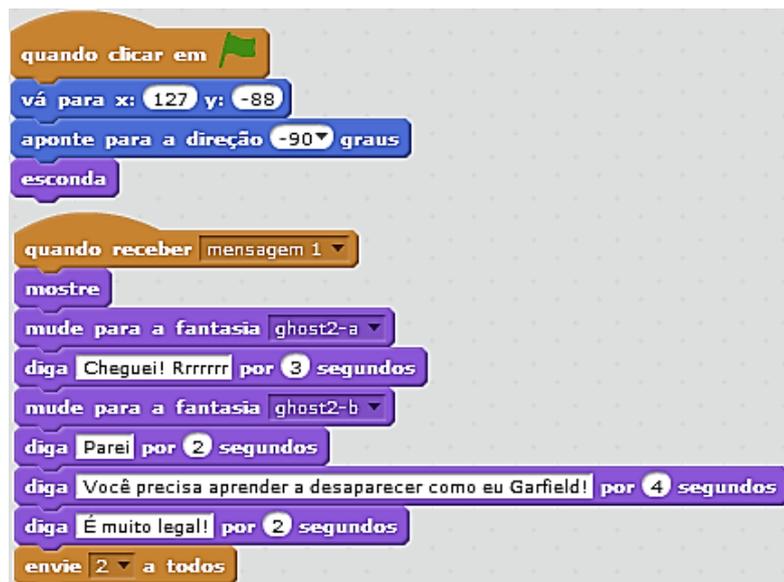
Nesta aula você utilizará 2 novos blocos: o  e o .

Eles são responsáveis pelo controle de tempo e aparição de atores e até mudança de panos de fundo. Escolha como atores o **GATO** e o **GHOST 2** e use pano de fundo **BADROOM 2**.

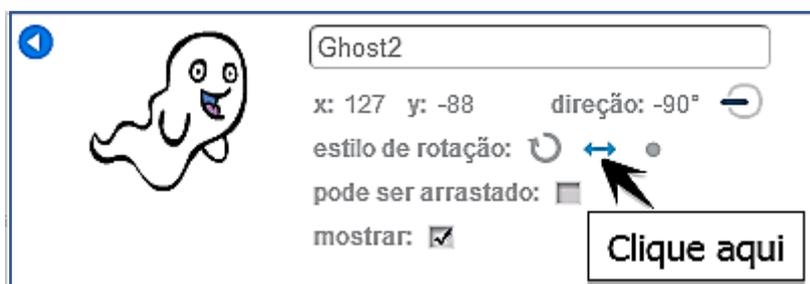
- Para o gato:



- Para o fantasma:



Clique na **bandeira**. Se seu fantasma aparecer virado para baixo, basta clicar no **i** e mudar o **estilo de rotação** dele, como indica na figura abaixo. Assim ele apontará para a esquerda (-90) mas não rotando e sim de forma espelhada.



Ficou claro qual é a função exata dos blocos de mensagens? A mensagem “avisa” o outro ator quando ele deve aparecer e vice-versa.

IMPORTANTE: As mensagens podem ser nomeadas como você quiser. O nome serve apenas para identificar no momento do envio e recebimento.

17ª AULA - PRATICANDO

Utilizando os conhecimentos que você adquiriu na aula de diálogos e na aula anterior sobre envio e recebimento de mensagens, abra o projeto feito na aula anterior e continue ele, adicionando mais 2 atores e mais 2 panos de fundo criando assim uma animação completa.

DESAFIO

Crie uma nova animação com o tema que você preferir com, no mínimo, 2 panos de fundo e 2 atores. Seja criativo!

18ª AULA – MATEMÁTICA NO SCRATCH 2.0

Como você sabe, a matemática está em todos os lugares. Para você criar jogos e animações, você utiliza matemática a todo momento sem saber. Os blocos que possuem operações matemáticas estão localizados na categoria **operadores**.

Hoje, você criará um game simples onde será aplicado algumas operações aritméticas básicas.

Escolha o pano de fundo **CHALKBOARD** e o ator **AVERY**. Escreva na lousa do pano de fundo a frase “**Aula de matemática**”. Seu pano de fundo deverá ficar assim:



IMPORTANTE: para escrever a frase utilize a fonte **GLÓRIA** e a cor que preferir. Essas configurações ficam na parte inferior do pano de fundo.

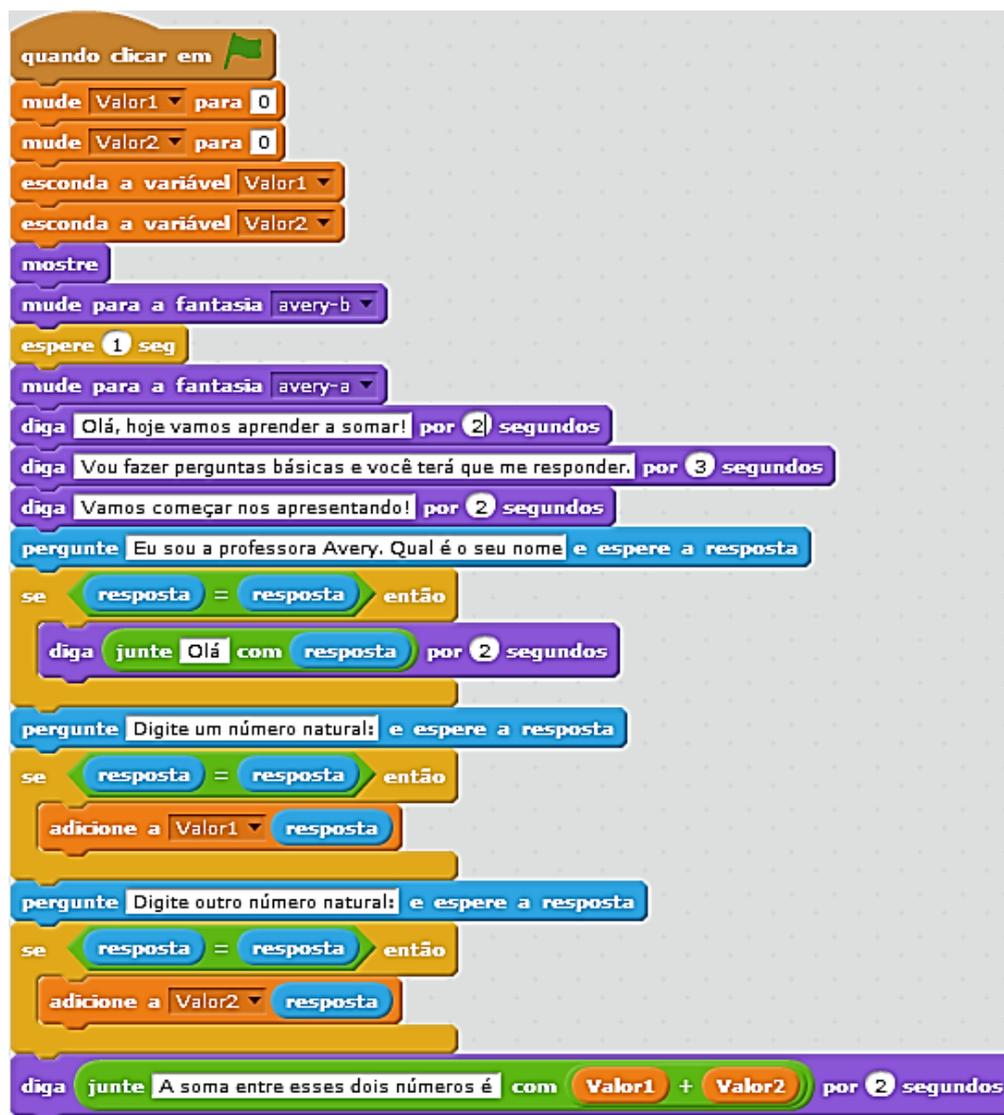
Crie 2 variáveis: uma chamada **valor 1** e outra **valor 2**. Há 2 partes da programação que talvez você não entenderá, sendo elas:



- A primeira é estruturada dessa maneira para o software não igualar sua resposta com qualquer outra coisa. Você fará exercício com estrutura diferente nas próximas aulas.

- A segunda estrutura **junta** um texto com o valor de uma soma.

Programa a Avery da seguinte maneira:



Não se assuste! É uma programação grande, porém não é tão difícil quanto parece ser. Tente interpretar cada passo e não apenas copiar.

Lembre-se: Para ser um bom programador é necessário muita dedicação e paciência para enfrentar as dificuldades.

Teste seu projeto e jogue à vontade.

19ª AULA - PRATICANDO

Utilizando o projeto da aula anterior, abra-o em seu computador e continue ele fazendo-o somar até 4 números. Em seguida, faça-o subtrair 4 números.

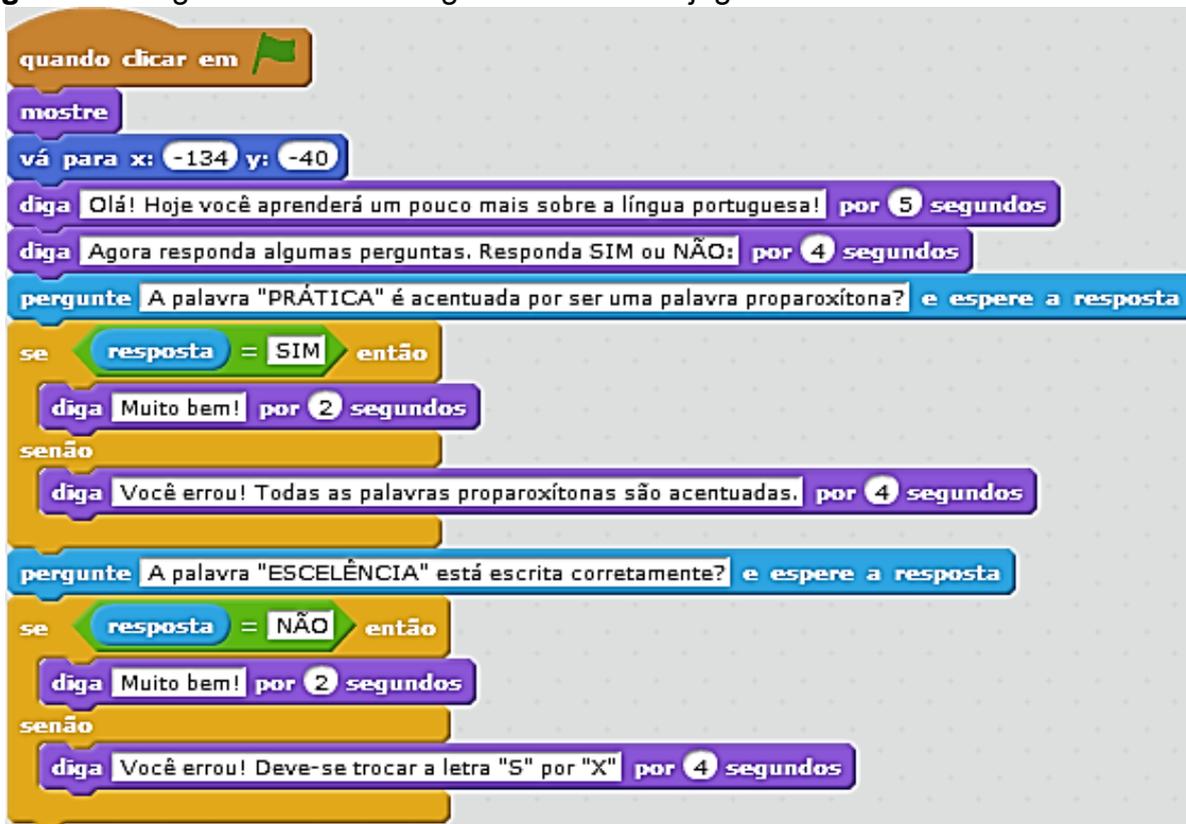
DESAFIO

Faça o game resolver as 4 operações aritméticas: adição, subtração, multiplicação e divisão.

20ª AULA – MEU PRIMEIRO QUIZ

Na última aula você aprendeu como se estrutura um jogo de perguntas e respostas e como utilizar as 4 operações aritméticas no **SCRATCH 2.0**. Hoje você irá aprender como se faz um Quiz relacionando agora a **língua portuguesa**.

A estrutura em si **não muda**. Apenas alguns blocos serão trocados ou excluídos. Escolha o mesmo pano de fundo e ator do exercício anterior e escreva na lousa “**língua portuguesa**”. Programe o ator da seguinte maneira e jogue uma vez:



Observe que neste quiz não foi utilizado variáveis, pois **não** foi necessário **armazenar** nenhuma resposta. A estrutura foi feita por **comparação**.

IMPORTANTE: Quando você digita **na programação** a resposta correta para fazer a comparação, **NUNCA** deixe espaço **antes ou depois** da palavra, pois o software entenderá como caractere.

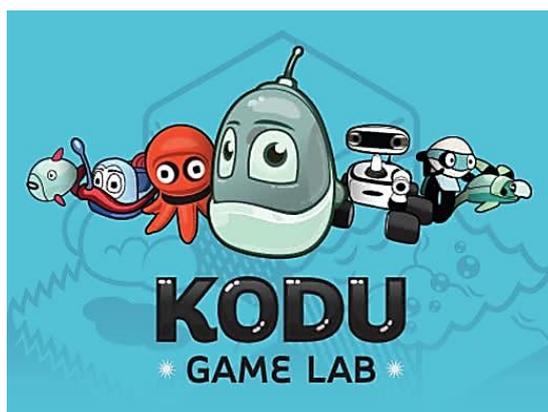
21ª AULA - PRATICANDO

Continue o quiz de língua portuguesa da aula anterior até chegar, no mínimo, 8 perguntas.

Em seguida, crie um placar de **acertos** e outro de **erros** e faça o quiz, ao atingir número máximo de pontos, dê como jogo vencido (Utilize mudança de pano de fundo, fantasias e o que mais achar necessário).

DESAFIO

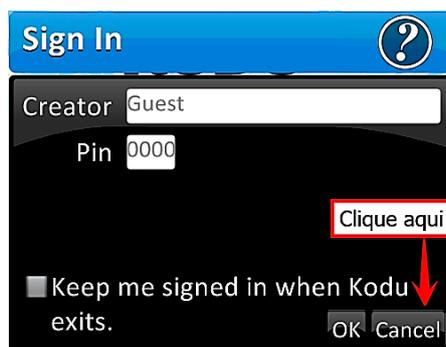
Faça um quiz contendo 3 personagens onde o primeiro fará 3 perguntas de matemática, o segundo fará 3 perguntas de língua portuguesa e o terceiro fará 3 perguntas de qualquer matéria de sua preferência. Somente quando o primeiro ator finalizar suas perguntas, o segundo deverá aparecer e assim sucessivamente.



O QUE É O KODU GAME LAB?

Kodu game lab é um ambiente de programação visual criado pela *microsoft* para que você possa criar os mundos e programar os personagens, desenvolvendo assim, jogos completos para se divertir durante as aulas.

Ao abrir o **kodu**, aguarde alguns segundos e a janela abaixo aparecerá na tela de seu computador:



Ao clicar em **cancel**, você vai encontrar o menu principal:



Clique em **novο mundo** para iniciar sua primeira aula.

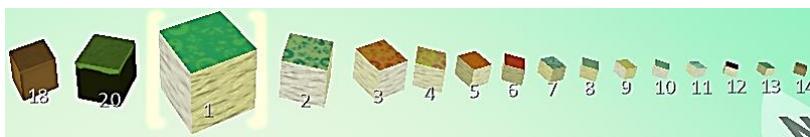
AULA 1 – CRIANDO MEU PRIMEIRO MUNDO



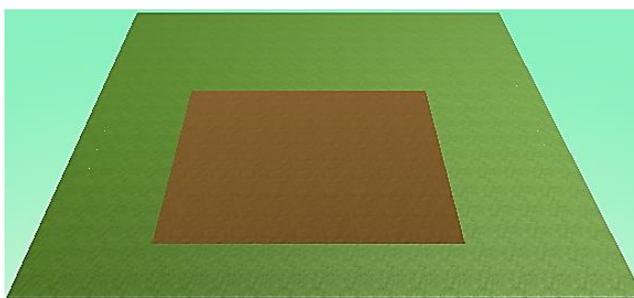
O quadrado **verde** é chamado de **terreno**, local onde você irá criar os mundos e programar todos os seus jogos. Abaixo do terreno, você encontra as ferramentas. você irá aprender a utilizar elas no decorrer das aulas.

Para iniciar a criação do seu mundo, clique na ferramenta .

Agora clique nas opções de terrenos . Ao clicar, aparecerá todos os terrenos para você escolher como mostra a figura abaixo:



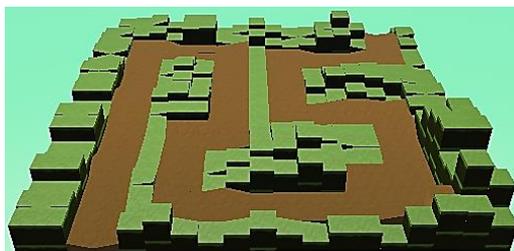
Para escolher os terrenos, utilize as setinhas direita  e esquerda  do teclado e, após escolher algum terreno, aperte o botão **enter**. Se você clicar no terreno verde, você irá ver que o quadrado de terra feito é muito grande, como mostra na figura a seguir.



AULA 2 – CRIANDO MONTES E VALES



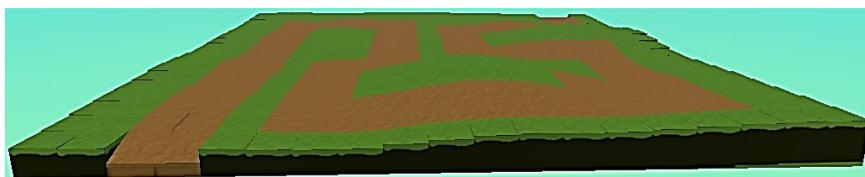
Você irá utilizar a ferramenta acima na aula de hoje. Ao clicar nela, basta clicar sobre o terreno e segurar o botão **esquerdo** do mouse. Na figura abaixo, mostra um terreno todo modificado por essa ferramenta:



Observe que os montes estão quadrados. Existe uma outra ferramenta que tem a função de **nivelar** esse terreno. É a ferramenta **suavizar terreno**.



Para usar ela, você fará o mesmo processo da construção dos montes, ou seja, clique na ferramenta e segure o botão **esquerdo** do mouse no local onde você quer modificar. O seu terreno deverá ficar, mais ou menos, dessa maneira:



Agora você vai **adicionar** algumas *árvores* nesse terreno. Para isto, você irá usar a ferramenta de objetos:



Clique nesta ferramenta e em seguida clique com o botão **direito** do mouse no terreno:

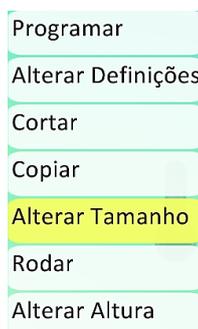


Agora clique em **adicionar objeto** para abrir o menu de opções, como mostra a figura abaixo:

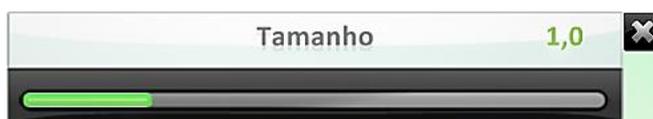


Clique na opção **tree**, que significa **árvore** em inglês. Você vai encontrar 4 tipos de árvores, então escolha uma delas, clique e posicione ela onde você preferir clicando no terreno.

Observação: a árvore ficou muito grande? Basta clicar com o botão direito sobre ela e selecionar a opção **alterar tamanho**:



Ao selecionar, abrirá a uma janela. Basta clicar sobre a barrinha verde e **arrastar** ela para **diminuir/aumentar** o tamanho da árvore:



DESAFIO

Abra um novo mundo e faça novos montes e vales, acrescente novas árvores da maneira que você quiser. Lembre-se que um mundo bem feito, fará seus jogos serem muito mais bonitos e legais de se jogar. Se divirta!

AULA 3 – MINHA PRIMEIRA PROGRAMAÇÃO

Hoje você irá fazer sua primeira programação. Você irá escolher um personagem e vai dar os comandos para ele fazer uma tarefa. Por exemplo, para uma pessoa atravessar uma avenida, existe um passo a passo como mostra abaixo:

PARE NA FAIXA DE PEDESTRES >> OLHE PARA A DIREITA >> OLHE PARA A ESQUERDA >> OLHE PARA FRENTE >> ANDE ATÉ A OUTRA CALÇADA.

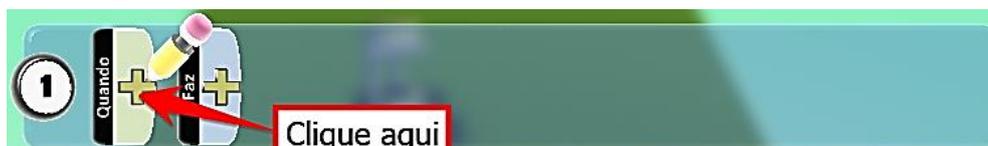
Viu como existe muitas tarefas na hora de atravessar uma avenida? Nos jogos funcionam da mesma maneira, é necessário colocar na programação tudo o que deve ser realizado. Abra um novo mundo e adicione o objeto **rover**, pois hoje você irá fazer ele se **movimentar**.



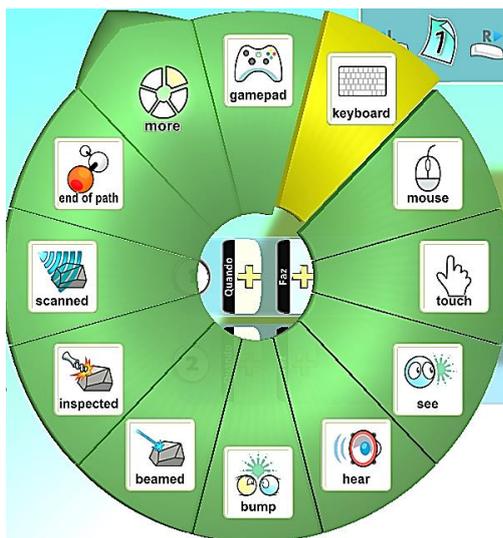
Clique com o botão direito sobre o personagem e selecione a opção **programar**.



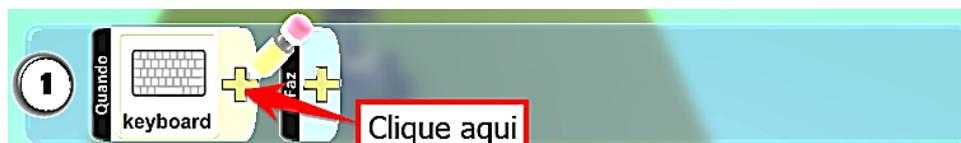
Ao clicar, a tela abaixo vai abrir no seu computador:



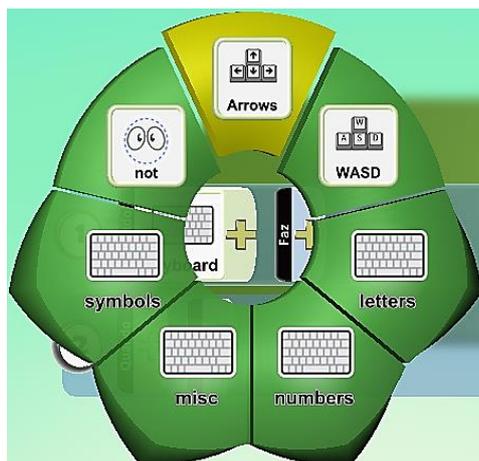
Em seguida, escolha a opção **keyboard**.



Ao escolher essa opção, você está “dizendo” para o personagem que vai usar o teclado. Agora é necessário informar quais teclas vão ser usadas para movimentar ele.



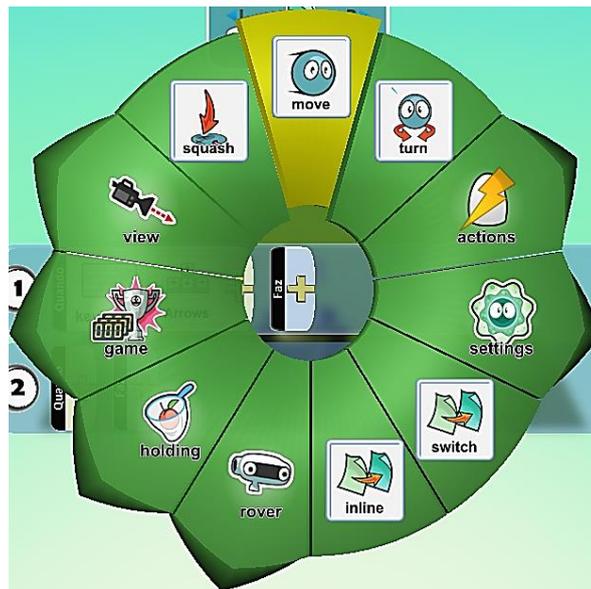
Escolha a opção **arrows**.



Agora que você já determinou quais botões vão ser usados, você vai dar o comando de **movimento**.



Após clicar onde está indicado na figura anterior, escolha a opção **move**.



Sua linha de programação deverá ficar dessa maneira:



Aperte 2 vezes o botão esc e movimente seu personagem conforme você programou, ou seja, pelas setinhas do teclado!

DESAFIO

Crie um mundo de sua preferência e faça o personagem se movimentar neste mundo através das teclas **w**, **a**, **s** e **d**.

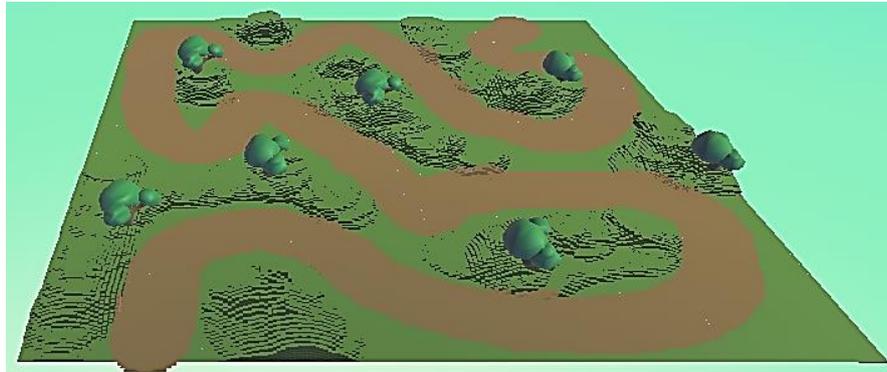
AULA 4 – CORRIDA DE MONOCICLOS

Hoje você e seus amigos vão elaborar um jogo de corrida de monociclos.

O que é um monociclo?

Monociclo é um velocípede de uma roda só e geralmente é usado em apresentações circenses (circos).

Elabore um mundo com montes, árvores e um percurso. A imagem abaixo, é um exemplo:



Em seguida, adicione 2 monociclos e posicione eles no início da pista como mostra a figura abaixo:



Mude as cores dos monociclos de acordo com sua preferência desde que fiquem com cores distintas. Para isso, basta colocar o cursor do mouse sobre o monociclo para aparecer as opções de cores:

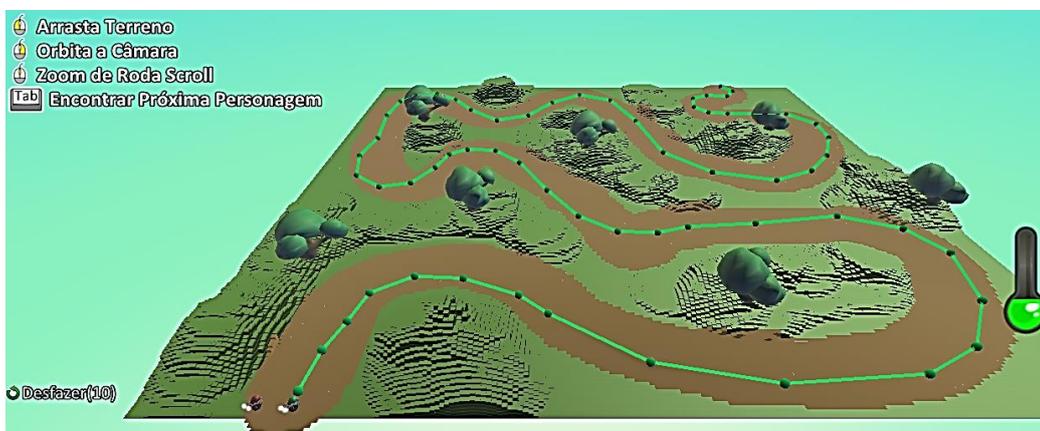


Usando as setinhas  , escolha a cor. Após a escolha, arraste o mouse para o lado mantendo a cor escolhida.

Em seguida você vai usar a ferramenta **adicionar trajeto**.



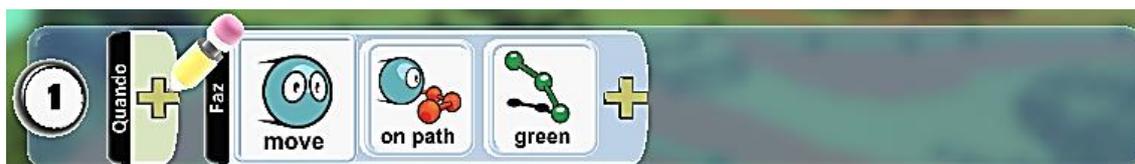
Na corrida, você vai controlar um monociclo e o outro vai correr sozinho, porém você precisa determinar onde esse monociclo vai passar, ou seja, vai ser necessário programar ele. Antes de iniciar a programação, com essa ferramenta você vai indicar o trajeto desse monociclo. Basta você ir clicando com o mouse e “esticando” as linhas na pista por onde você quer que ele passe. Ao finalizar, coloque a cor desse trajeto traçado na mesma cor que o monociclo (para mudar a cor, você vai usar o mesmo método usado no monociclo), como mostra a figura a seguir:



Agora você vai iniciar a programação. Programe o **seu** monociclo dessa maneira:



Programe o outro monociclo assim:



Essa última programação diz o seguinte: **“mova na direção da linha verde”**. Feito as programações, aperte esc e se divirta com seu game de corrida!

DESAFIO

Adicione mais monociclos e faça uma grande corrida para se divertir com seus amigos.

AULA 5 – PAC-MAN

Na aula de hoje você e seus amigos vão construir um game muito famoso, o *pac-man*. Ao abrir um novo mundo, aumente o tamanho dele. Em seguida, adicione 2 personagens *kodu* em seu mundo com cores distintas.



Agora você deve programar cada personagem. A programação do **kodu vermelho** vai ficar assim:



A primeira linha de programação é a de **movimentação** e ela diz o seguinte: **“quando as setas do teclado forem pressionadas, faça mover”**.

A segunda linha é a programação para o kodu comer as maçãs que estiverem no caminho e ela diz o seguinte: **“quando encostar nas maçãs, faça comer elas”**.

Agora você irá programar o **kodu azul**, e ele tem a função de perseguir o **kodu vermelho**. A programação dele deve ser assim:



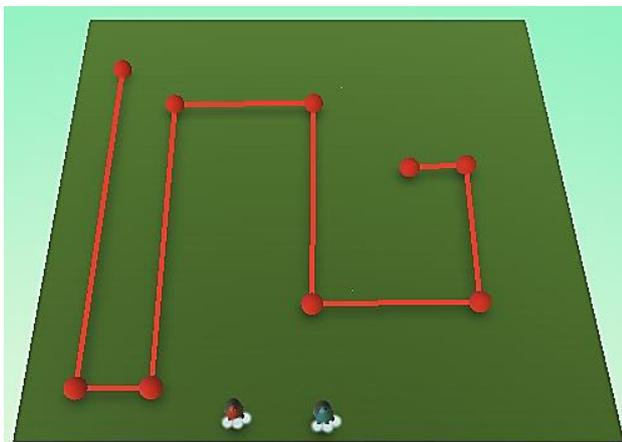
A primeira linha de programação diz o seguinte: **“quando ver o kodu vermelho, faça mover na direção dele”**.

A segunda diz: **“quando encostar no kodu vermelho, faça comer ele”**.

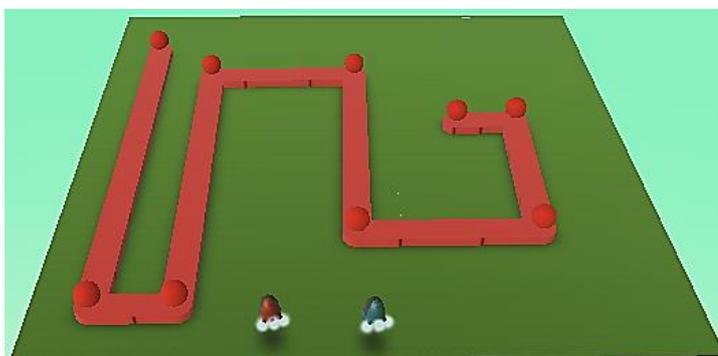
A terceira diz que depois da execução da linha 2, será fim de jogo. Por esse motivo ela **deve** estar **“arrastada”** um pouco para direita.

Agora adicione algumas maçãs no seu mundo, aperte o botão **ESC** e teste seu game comendo as maçãs. Agora o próximo passo é criar os muros do labirinto e para isto você vai usar a mesma ferramenta de criar trajeto usada na aula anterior.

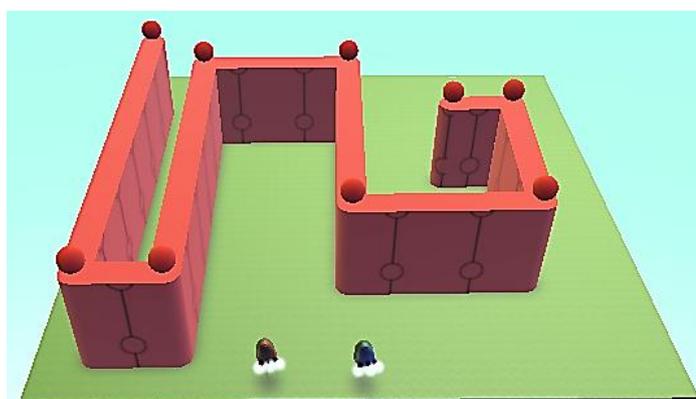
Trace as linhas onde serão as paredes do seu labirinto como mostra abaixo:



Feito as linhas, basta segurar o botão **SHIFT** e usar as setas do teclado para cima e baixo para escolher o tipo de trajeto que, no caso, vai ser uma parede.



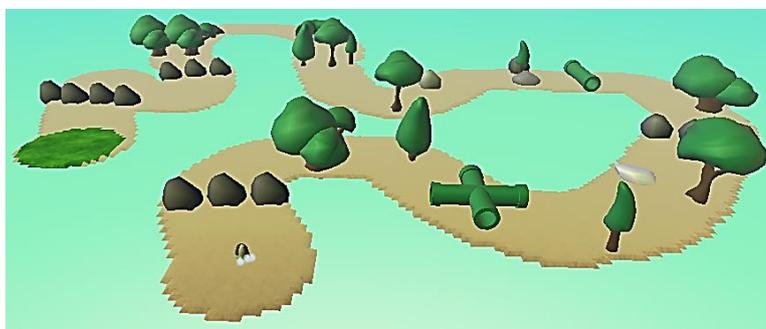
Agora segure **SHIFT** novamente e clique com o botão direito do mouse em cima da parede e selecione **alterar altura**. Aumente a parede na altura que você preferir.



Agora é só se divertir com seus amigos!

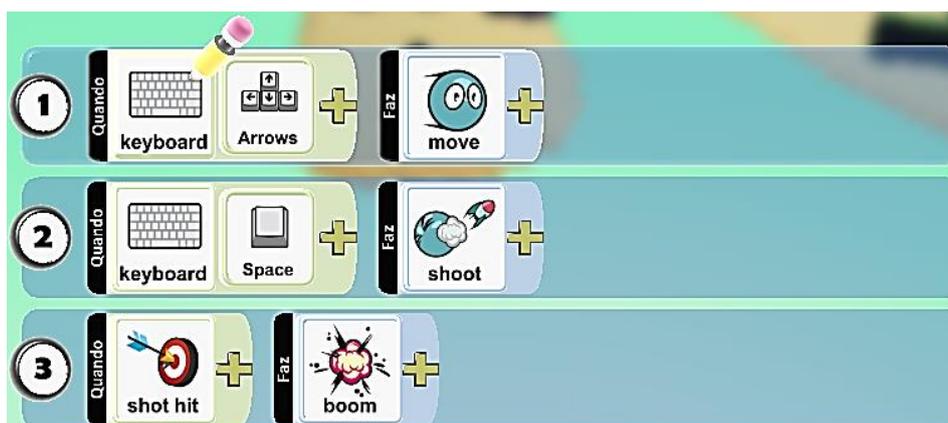
AULA 6 – TIRO AO ALVO

Você já deve ter jogado algum game de tiro ao alvo, certo? Senão, chegou a hora de você não só jogar, mas criar o seu próprio game de tiro ao alvo. Você vai criar um mundo de sua preferência, mas é necessário ter um percurso, um **kodu** e muitas pedras e árvores ao longo desse percurso, pois você irá destruir elas para chegar ao final do caminho. A imagem abaixo é um exemplo de mundo:



O objetivo do game é chegar no final do percurso, ou seja, será preciso abrir o caminho até lá. As pedras, árvores e outros objetos, como mostra na figura acima, são obstáculos e você vai precisar eliminá-los do percurso.

para isto, você terá que programar o **kodu** para ele se **movimentar** e para **atirar** nos obstáculos. Então programe o **kodu** como mostra abaixo:



A primeira linha de programação você já conhece, é para movimentação. Ela diz o seguinte: **“quando as setas do teclado forem pressionadas, faça mover”**.

a segunda linha é a programação para **atirar** nos obstáculos e será usado o botão **espaço** para essa função. Essa linha diz o seguinte: **“quando botão espaço do teclado for pressionado, faça atirar”**.

A terceira linha será para quando o tiro acertar um objeto, ele explodir e desaparecer. Ela diz o seguinte: **“quando um alvo for atingido, faça explodir”**. Quais são os alvos? São os obstáculos que você vai encontrar no caminho.

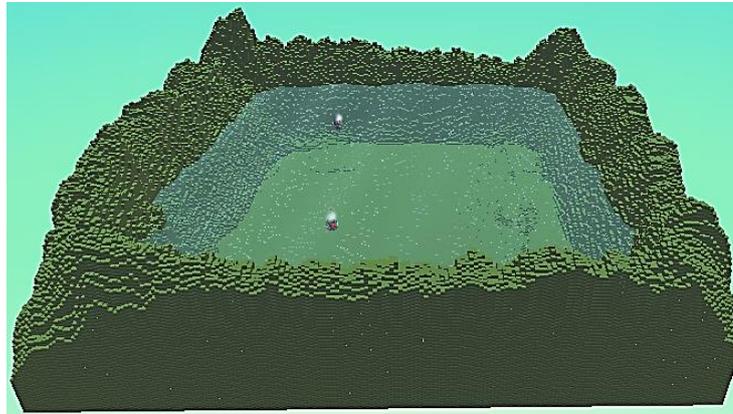
Agora se divirta com seu game de tiro ao alvo.

DESAFIO

Adicione maçãs no seu percurso e faça o **kodu** comer elas.

AULA 7 – BATALHA MARINHA

Na aula de hoje você vai fazer um game de batalha marinha, onde você vai controlar um navio e terá que eliminar os outros, pois eles vão te atacar. Abra um novo mundo e aumente o tamanho dele. Agora crie um mundo como mostra abaixo:



Para adicionar água, basta você clicar na ferramenta de água . Essa ferramenta é usada da mesma maneira quando se quer adicionar montes e vales, ou seja, clique e mantenha pressionado o botão direito do mouse até encher de água como mostra na figura acima. Em seguida adicione um barco como personagem:



Programa esse barco da seguinte maneira:



A programação anterior diz o seguinte:

Linha 1: “quando a setas do teclado forem pressionadas, faça mover”.

Linha 2: “quando o botão espaço do teclado for pressionado, faça atirar”.

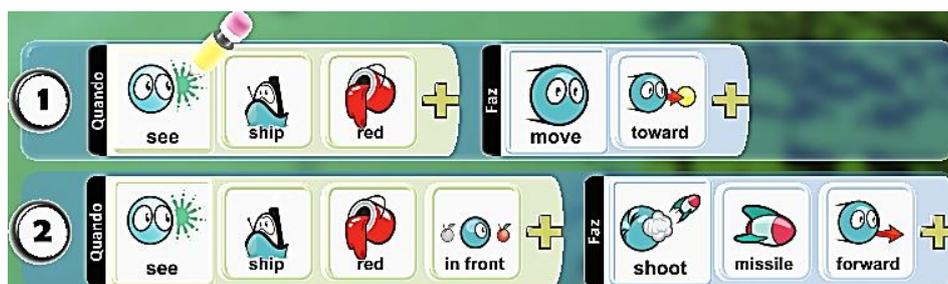
Linha 3: “quando um alvo for atingido, faça explodir”.

Linha 4: “quando um alvo for atingido, faça adicionar pontos no placar vermelho”.

Linha 5: “quando o placar vermelho for igual a 10 pontos, faça vencer o jogo”.

OBSERVAÇÃO: Pinte o seu barco da cor **vermelha**.

Agora você vai programar o outro barco para ele te perseguir. Programe ele como mostra abaixo:

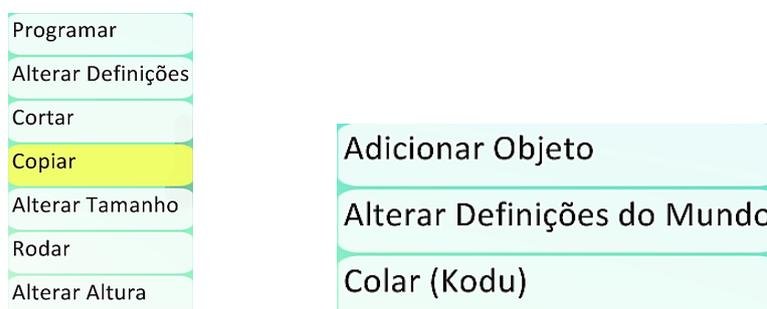


Esta programação diz o seguinte:

Linha 1: “quando ver o barco vermelho, faça mover na direção dele”.

Linha 2: “quando ver o barco vermelho frontalmente, faça atirar míssil em linha reta”.

agora teste seu game. Em seguida, duplique 10 vezes o barco, clicando com o botão direito do mouse nele e selecionando a opção **copiar**. É feito o mesmo processo para **colar** o barco copiado:



Divirta-se com seus amigos e faça uma grande batalha!

DESAFIO

Acrescente peixes e submarinos e façam eles se moverem aleatoriamente na água.

DESAFIOS EXTRAS

DESAFIO 8

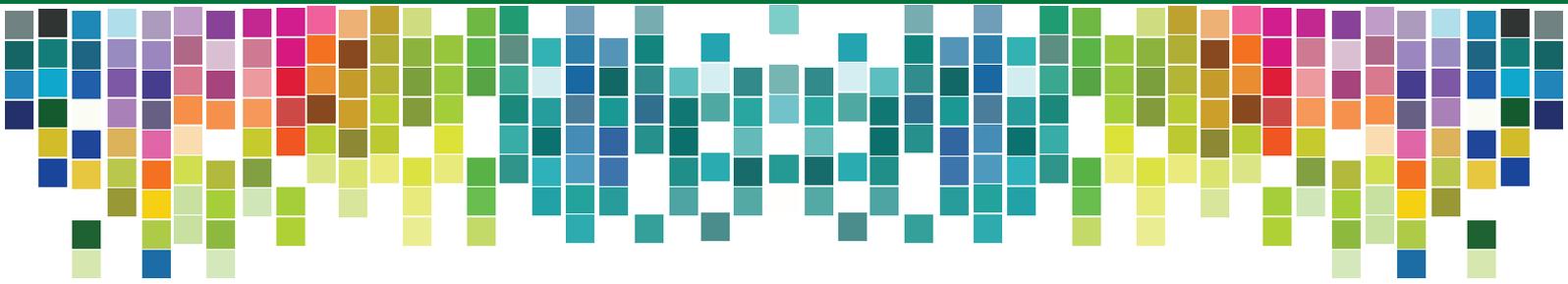
Elabore um mundo aquático como na aula anterior e faça um submarino coletar estrelas do mar. ao coletar 20 estrelas, dê como jogo vencido.

DESAFIO 9

Faça uma corrida de peixes, onde você terá que criar os percursos e obstáculos por onde eles deverão passar.

AULA LIVRE

Você chegou na tão esperada aula livre! hoje você vai poder criar o game que você quiser e se divertir à vontade com seus amigos. use sua criatividade e elabore um game fantástico no **kodu!**



KIDS MAKER

NOME: _____

TURMA: _____ HORÁRIO: _____

DIA DA SEMANA: _____



 **Jovem Engenheiro**
ESCOLA DE NOVAS TECNOLOGIAS

