

MIT
APP INVENTOR

Crie aplicativos!

- **Sobre**
[Sobre o App Inventor](#) [Nosso time](#) [Treinadores especialistas](#) [Aplicativo do mês](#) [Appathon](#) [Termos de serviço](#) [Notas de lançamento](#)
- **Educadores**
[Ensinar](#) [Tutoriais](#) [IA com App Inventor](#)
- **Notícias**
[Nas notícias](#) [Eventos](#) [Histórias do campo](#)
- **Recursos**
[Fundação App Inventor](#) [Iniciar](#) [Documentação](#) [Suporte e comunidade](#) [Tutoriais](#) [livros](#) [Informações de Código Aberto](#) [Pesquisar](#) [Hora do Código](#) [Recursos adicionais](#)
- **Blogues**
[Blog do Inventor de Aplicativos](#)

Doar

Mole Mash

No jogo **MoleMash**, uma toupeira aparece em posições aleatórias em um campo de jogo, e o jogador marca pontos acertando a toupeira antes que ela pule para longe. Este tutorial mostra como construir o **MoleMash** como exemplo de um jogo simples que usa animação. [Veja o capítulo do livro do App Inventor Book de Wolber, et al. \(Inclui tutorial em VÍDEO\)](#)

Começando

Conecte-se ao site do App Inventor e inicie um novo projeto. Nomeie-o como "MoleMash" e também defina o **título** da tela como "MoleMash". Abra o Blocks Editor e conecte-se ao telefone.

Baixe também esta [imagem de uma toupeira](#) e salve-a em seu computador.

Introdução

Você projetará o jogo para que a toupeira se mova uma vez a cada meio segundo. Se for tocado, a pontuação aumenta em um e o telefone vibra. Pressionar reiniciar redefine a pontuação para zero.

Este tutorial apresenta:

- sprites de imagem
- temporizadores e o componente **Relógio**
- procedimentos
- escolher números aleatórios entre 0 e 1
- blocos de texto
- bloqueio de tipo

Os primeiros componentes

Vários componentes devem ser familiares dos tutoriais anteriores:

- Uma **tela** chamada "MyCanvas". Esta é a área onde a toupeira se move.
- Uma **etiqueta** chamada "ScoreLabel" que mostra a pontuação, ou seja, o número de vezes que o jogador acertou a toupeira.
- Um **botão** chamado "ResetButton".

Arraste esses componentes da Paleta para o Visualizador e atribua seus nomes. Coloque **MyCanvas** no topo e defina suas dimensões para 300 pixels de largura por 300 pixels de altura. Defina o **Texto** de ScoreLabel como "Pontuação: ---". Defina o **texto** do ResetButton como "Redefinir". Adicione também um componente **de som** e nomeie-o como "Ruído". Você usará **Noise** para fazer o telefone vibrar quando a toupeira for atingida, semelhante à maneira como você fez o gatinho ronronar em **HelloPurr**.

Temporizadores e o componente Relógio



Você precisa fazer com que a toupeira pule periodicamente e fará isso com a ajuda de um componente **Relógio**. O componente Clock fornece várias operações relacionadas ao tempo, como informar a data. Aqui, você usará o componente como um temporizador que dispara em intervalos regulares. O intervalo de disparo é determinado pela propriedade **TimerInterval** do Clock. Arraste um componente Relógio; ele irá para a área de componentes não visíveis. Nomeie-o "MoleTimer". Defina seu **TimerInterval** para 500 milissegundos para fazer a toupeira se mover a cada meio segundo. Certifique-se de que **TimerEnabled** esteja marcado.

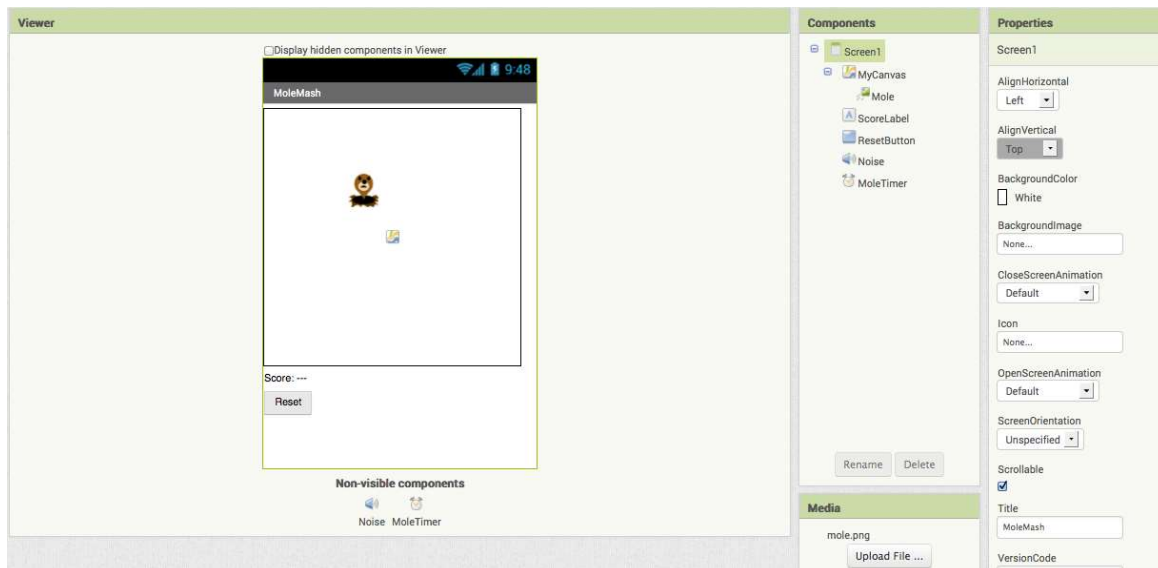
Adicionando um Sprite de Imagem

Para adicionar a toupeira em movimento, usaremos um *sprite*.

Sprites são imagens que podem se mover na tela dentro de um Canvas. Cada sprite tem uma **Speed** e um **Heading**, e também um **Interval** que determina com que frequência o sprite se move em sua velocidade designada. Sprites também podem detectar quando são tocados. No **MoleMash**, a toupeira tem velocidade zero, portanto não se moverá sozinha. Em vez disso, você definirá a posição da toupeira toda vez que o cronômetro disparar. Arraste um componente **ImageSprite** para o Viewer. Você encontrará esse componente na categoria Desenho e Animação da Paleta. Coloque-o na área **MyCanvas**. Defina estas propriedades para o sprite Mole:

- **Imagem**: Use mole.png, que você baixou para o seu computador no início deste tutorial.
- **Ativado**: marcado
- **Intervalo**: 500 (O intervalo não importa aqui, porque a velocidade da toupeira é zero: ela não está se movendo sozinha.)
- **Rumo**: 0 O rumo também não importa aqui, porque a velocidade é 0.
- **Velocidade**: 0,0
- **Visível**: marcado
- **Largura**: Automático
- **Altura**: Automático

Você deve ver as propriedades **x** e **y** já preenchidas. Elas foram determinadas por onde você colocou a toupeira quando a arrastou para o **MyCanvas**. Vá em frente e arraste a toupeira um pouco mais. Você deve ver a mudança de **x** e **y**. Você também deve ver a toupeira em seu telefone conectado e a toupeira se movendo no telefone conforme você a arrasta no Designer. Agora você especificou todos os componentes. O Designer deve ficar assim. Observe como **Mole** é recuado em **MyCanvas** na lista de estrutura de componentes, indicando que o sprite é um subcomponente da tela.



Comportamento de componentes e manipuladores de eventos

Agora você especificará o comportamento do componente. Isso apresenta algumas novas ideias do App Inventor. A primeira é a ideia de um *procedimento*. Para uma visão geral e explicação dos procedimentos, confira a [página Procedimentos](#).

Um procedimento é uma sequência de instruções às quais você pode se referir todas de uma vez como um único comando. Se você tem uma sequência que precisa usar mais de uma vez em um programa, pode defini-la como um procedimento e não precisa repetir a sequência toda vez que usá-la. Os procedimentos no App Inventor podem receber argumentos e valores de retorno. Este tutorial cobre apenas o caso mais simples: procedimentos que não recebem argumentos e não retornam valores.

Definir procedimentos

Defina dois procedimentos:

- **MoveMole**: move o sprite **Mole** para uma nova posição aleatória na tela.
- **UpdateScore**: mostra a pontuação, alterando o texto do **ScoreLabel**

Comece com o MoveMole:

- No Blocks Editor, em Built-In, abra a gaveta de Procedimentos. Arraste um bloco **de procedimento** e altere o rótulo "procedimento" para "MoveMole".

Nota: Existem dois blocos semelhantes: **to procedure do** e **to procedure result**. Aqui você deve usar **o procedimento do**.

O bloco **to MoveMole** tem um slot chamado "do". É aí que você coloca as declarações para o procedimento. Nesse caso, haverá duas instruções: uma para definir a posição x da toupeira e outra para definir sua posição y. Em cada caso, você definirá a posição como uma fração aleatória, entre 0 e 1, da diferença entre o tamanho da tela e o tamanho da toupeira.

Você cria esse valor usando blocos para `fração aleatória` e multiplicação e subtração. Você pode encontrá-los na gaveta de Matemática.

- Crie o procedimento `MoveMole`. A definição completa deve ficar assim:



`MoveMole` não aceita nenhum argumento, então você não precisa usar a função `mutator` do bloco de procedimento. Observe como os blocos se conectam: a primeira instrução usa o bloco `set Mole.X` para definir a posição horizontal da toupeira. O valor inserido na tomada do bloco é o resultado da multiplicação:

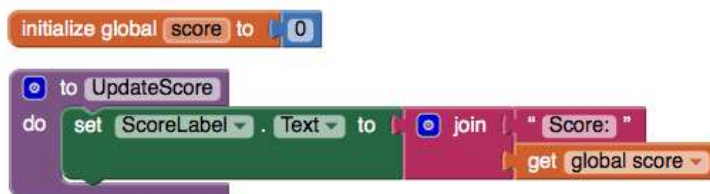
1. O resultado da `chamada` do bloco de fração aleatória, cujo valor entre 0 e 1
2. O resultado da subtração da largura da toupeira da largura da tela

A posição vertical é tratada de forma semelhante.

Com o `MoveMole` pronto, o próximo passo é definir uma variável chamada `score` para armazenar o score (número de acertos) e dar a ele o valor inicial 0. Defina também um procedimento `UpdateScore` que mostre o score em `ScoreLabel`. O conteúdo real a ser mostrado no `ScoreLabel` será o texto "Score:" junto ao valor de `score`.

- Para criar a parte "Pontuação:" do rótulo, arraste um bloco de texto da gaveta Texto. Altere o bloco para ler "Pontuação:" em vez de "".
- Use um bloco de junção para anexá-lo a um bloco que fornece o valor da variável de pontuação. Você pode encontrar o bloco de junção na gaveta Texto.

Veja como a `pontuação` e o `UpdateScore` devem ficar:



Adicionar um temporizador

O próximo passo é fazer com que a toupeira continue se movendo. Aqui é onde você usará o `MoleTimer`. Os componentes do relógio têm um manipulador de eventos chamado `when ... Timer` que dispara repetidamente em uma taxa determinada pelo `TimerInterval`.

Configure o `MoleTimer` para chamar o `MoveMole` toda vez que o cronômetro disparar, criando o manipulador de eventos como este:



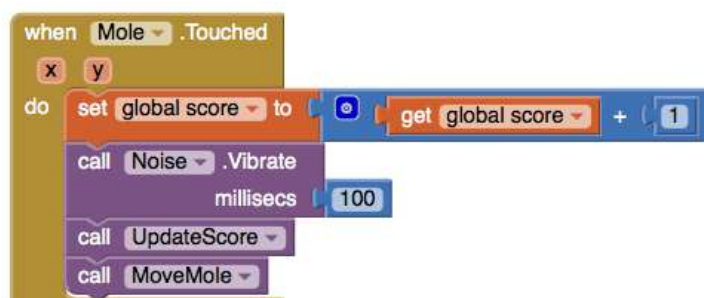
Observe como a toupeira começa a pular no telefone assim que você define o manipulador de eventos. Este

Adicione um manipulador de toque de toupeira

O programa deve aumentar a pontuação cada vez que a toupeira é tocada. Sprites, como telas, respondem a eventos de toque. Portanto, crie um manipulador de eventos de toque para o `Mole` que:

1. Incrementa a pontuação.
2. Chama `UpdateScore` para mostrar a nova pontuação.
3. Faz o telefone vibrar por 1/10 de segundo (100 milissegundos).
4. Chama `MoveMole` para que a toupeira se mova imediatamente, em vez de esperar pelo cronômetro.

Veja como isso se parece em blocos. Vá em frente e monte os blocos `when Mole.Touched` como mostrado.



Aqui vai uma dica: Você pode usar [typeblocking](#) : digitando para criar blocos rapidamente.

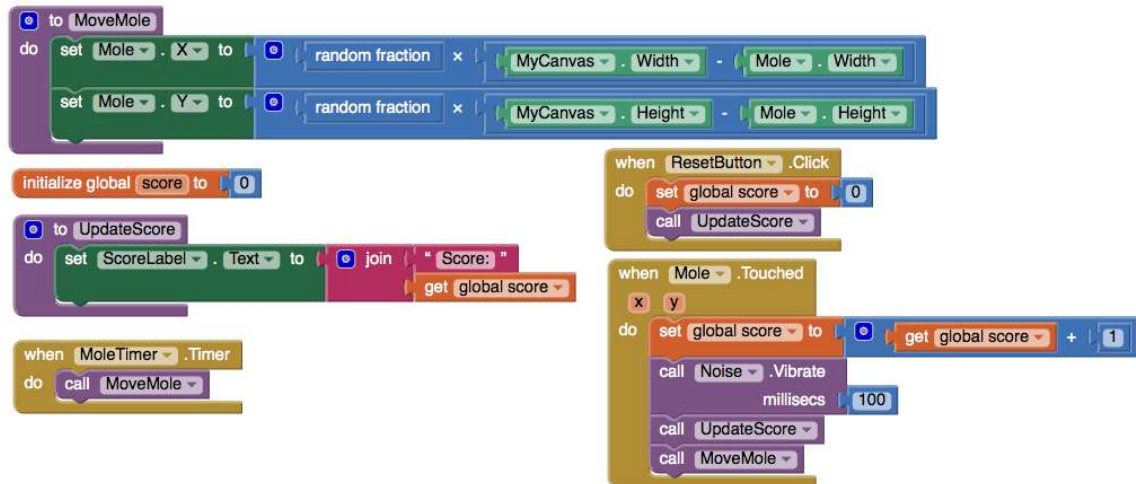
- Para criar um bloco de valor contendo 100, basta digitar 100 e pressionar Enter.
- Para criar um bloco `MoveMole`, basta digitar `MoveMole` e selecionar o bloco desejado na lista

Redefinir a pontuação

Um detalhe final é zerar a pontuação. Isso é simplesmente uma questão de fazer o **ResetButton** alterar a pontuação para 0 e chamar `UpdateScore`.

programa completo

Aqui está o programa **MoleMash** completo :



variações

Depois de fazer o jogo funcionar, você pode querer explorar algumas variações. Por exemplo:

- Faça o jogo variar a velocidade da toupeira em resposta ao desempenho do jogador. Para variar a rapidez com que a toupeira se move, você precisará alterar a propriedade **Interval** do **MoleTimer**.
- Acompanhe quando o jogador acerta a toupeira e quando o jogador erra a toupeira e mostra uma pontuação com acertos e erros. Para fazer isso, você precisará definir manipuladores tocados para **Mole**, como agora, e para **MyCanvas**. Uma questão sutil, se o jogador tocar na toupeira, isso também conta como um toque para o **MyCanvas**? A resposta é sim. Ambos os eventos de toque serão registrados.

Análise

Aqui estão algumas das ideias abordadas neste projeto:

- Sprites são formas sensíveis ao toque que você pode programar para se mover em um **Canvas**.
- O componente **Relógio** pode ser usado como um **cronômetro** para criar eventos que acontecem em intervalos regulares.
- Procedimentos são definidos usando blocos `de procedimentos`.
- Para cada procedimento definido, o App Inventor cria automaticamente um bloco de chamada associado e o coloca na gaveta Procedimentos.
- Fazer um bloco `de fração aleatória` produz um número entre 0 e 1.
- Os blocos de texto especificam texto literal, semelhante à maneira como os blocos numéricos especificam números literais.
- [Typeblocking](#) é uma maneira de criar blocos rapidamente, digitando o nome de um bloco.

Digitalize o aplicativo de exemplo para o seu telefone

Digitalize o código de barras a seguir em seu telefone para instalar e executar o aplicativo de amostra.



Baixar código-fonte

Se você gostaria de trabalhar com esta amostra no App Inventor, baixe o [código-fonte](#) para o seu computador, abra o App Inventor, clique em **Projetos**, escolha **Importar projeto (.aia) do meu computador...**, e selecione o código-fonte que você acabou de baixar.

Feito com MoleMash? Retorne aos outros tutoriais [aqui](#).

Inventor de aplicativos do MIT



© 2012-2022 Instituto de Tecnologia de Massachusetts



Este trabalho está licenciado sob uma Licença Creative Commons Atribuição-Compartilhagual 4.0 Internacional .
Termos de serviço e política de privacidade

Suporte do App Inventor: Comunidade
Outras dúvidas: E-mail
GitHub: mit-cml
Acessibilidade: [acessibilidade.mit.edu](https://www.mit.edu/accessibilidade)



Texto original

Sugerir uma tradução melhor