Mole Mash V2 com

Sprite Layering

O que você está construindo

Este tutorial mostra como construir um jogo semelhante ao jogo de fliperama Whac-A-Mole™. O objetivo é tocar em uma toupeira que sai aleatoriamente de um dos cinco orifícios fixos. Cada vez que você consegue, sua pontuação aumenta em um ponto.

Para destacar os recursos do App Inventor **Any Component** e ImageSprite **Z-layering** , este aplicativo adota uma abordagem ligeiramente diferente do tutorial original do Mole Mash , que você **não precisa** ter concluído para concluir este. Você deve, no entanto, estar familiarizado com os fundamentos do App Inventor, usando o Designer para construir uma interface de usuário e usando o Blocks Editor para codificar o comportamento de um aplicativo. Se você não estiver familiarizado com o básico, tente percorrer alguns dos tutoriais básicos antes de continuar.

Introdução

Este tutorial inclui:

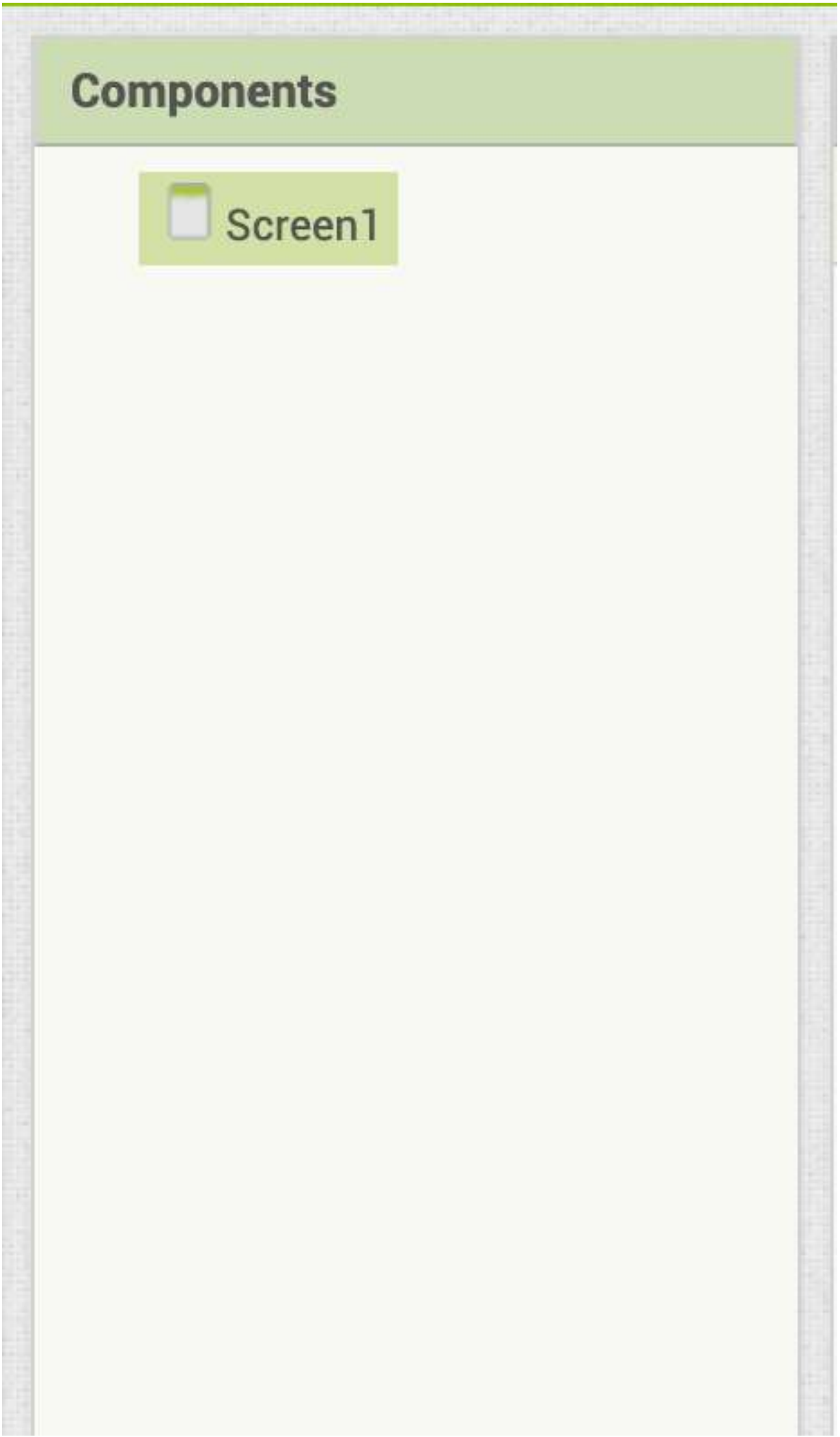
1. Adicionando componentes no Designer
2. Usando a funcionalidade *Any Component* para obter e definir propriedades de componentes ImageSprite
3. Controlando o jogo com o componente **Relógio**

4. Usando Sprite *Z-layering* para garantir que um ImageSprite apareça na frente do outro

Começando

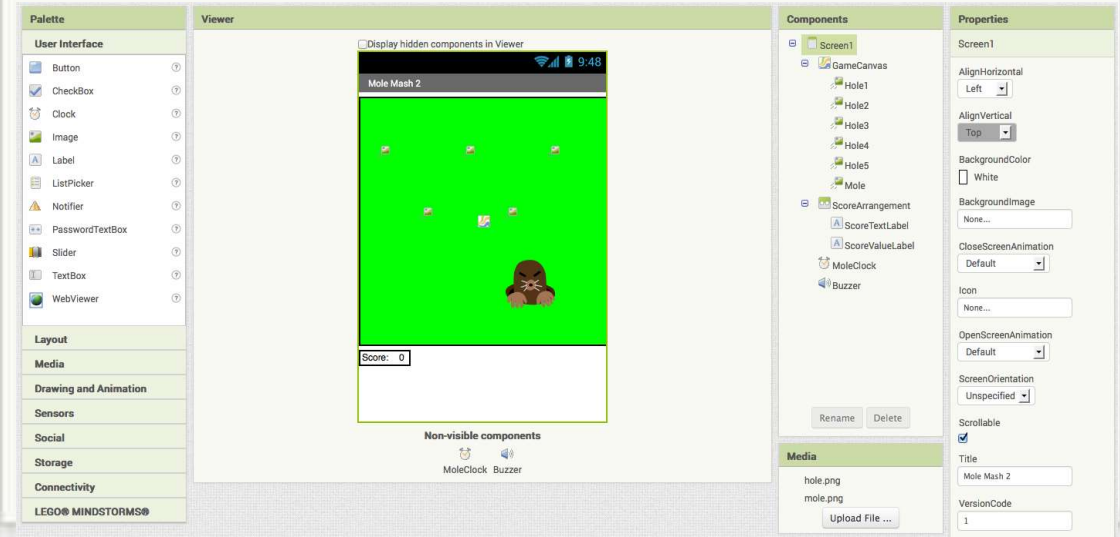
Abra ai2.appinventor.mit.edu e inicie um novo projeto. Defina a propriedade *Title* da tela com um nome apropriado, como "Mole Mash". Baixe os arquivos de imagem abaixo (criados por Yun Miao) para o seu computador clicando com o botão direito do mouse neles e, em seguida, adicione-os ao seu projeto pressionando o botão "Carregar arquivo..." no painel Mídia.

Buraco: Toupeira: 


Components
Screen1

Configure os componentes

A interface do usuário conterá um total de 6 ImageSprites: 5 buracos imóveis e 1 toupeira. A toupeira se moverá em cima dos buracos. Use o Designer para criar a interface do usuário. Quando terminar, deve ficar algo parecido com a imagem abaixo. Não se preocupe em alinhar os buracos uniformemente. Você especificará suas localizações por meio de suas propriedades *X* e *Y*. Instruções adicionais estão abaixo da imagem.



A seguir está a lista de componentes que você adicionará:

Tipo de componente	Grupo de Paletas	O que você vai nomear	Finalidade do Componente
Tela	Desenho e Animação	GameCanvasGenericName	o campo de jogo
ImageSprite (5)	Desenho e Animação	Buraco1 ... Buraco5	Buracos de onde a toupeira pode aparecer
ImageSpriteName	Desenho e Animação	Verruga	A toupeira
Disposição Horizontal	Disposição	ScoreArrangement	Para exibir a pontuação
Rótulo	Interface de usuário	ScoreTextLabel	Para segurar "Pontuação: "
Rótulo	Interface de usuário	ScoreValueLabel	Para segurar a pontuação (número de vezes que a toupeira foi atingida)
Relógio	Interface de usuário	MoleClock	Para controlar o movimento da toupeira
Som	meios de comunicação	campainha	Para vibrar quando a toupeira é tocada

Faça as seguintes alterações nas propriedades dos componentes:

Componente	Ação
Tela1	Defina *BackgroundColor* como Verde. Defina *Largura* para 320 pixels.

	Defina *Altura* para 320 pixels.
Buraco1	Defina *X* como 20 e *Y* como 60 (canto superior esquerdo).
Buraco2	Defina *X* para 130 e *Y* para 60 (centro superior).
Buraco3	Defina *X* para 240 e *Y* para 60 (canto superior direito)
Buraco4	Defina *X* como 75 e *Y* como 140 (canto inferior esquerdo).
Buraco5	Defina *X* para 185 e *Y* para 140 (canto inferior direito).
Verruga	Defina *a imagem* como "mole.png". Defina *Z* como 2 para que a toupeira apareça na frente dos outros *ImageSprites , que têm o valor Z* padrão de 1.
ScoreTextLabel	Defina *o texto* como "Pontuação: ".
ScoreTextValue	Defina *o texto* como "0".

Não se preocupe agora em definir a propriedade *Picture* para os furos; definiremos a propriedade no Editor de Blocos.

Adicionar comportamentos aos componentes

Aqui está uma visão geral do que precisamos criar blocos para fazer:

1. Criar variável global:
   - buracos : uma lista de buracos ImageSprites
2. Quando o aplicativo é iniciado:
   - Preencha a lista de furos.
   - Defina a propriedade *Picture* de cada furo como "hole.png".
   - Mova a toupeira aleatoriamente para um buraco.
3. Crie um procedimento MoveMole para:
   - Defina a variável local currentHole para um buraco aleatório da lista de buracos .
   - Mova a toupeira para o local de currentHole .
4. Quando o MoleClock.Timer é acionado:
   - Chame MoveMole para mover a toupeira aleatoriamente.
5. Implemente um manipulador que faz o seguinte quando a toupeira é tocada:
   - Adicione um à pontuação.
   - Faça o telefone vibrar brevemente.
   - Ligue para a MoveMole .

Para continuar, mude para o Editor de Blocos.

Criando Variáveis

Crie a variável e nomeie-a como hole . Por enquanto, daremos a ela um valor inicial "fictício" de uma lista vazia. Definiremos o valor inicial real no manipulador de eventos Screen1.Initialize , que é executado toda vez que o aplicativo carrega a tela. Aqui está uma foto e uma lista dos blocos que você vai precisar:

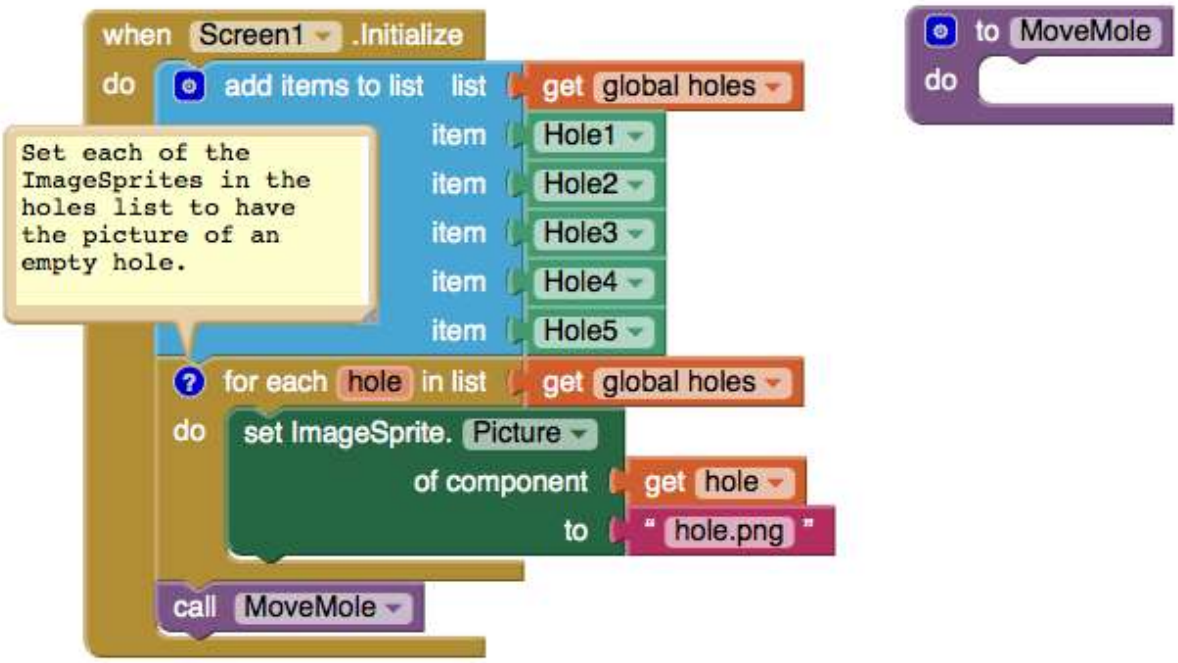
Tipo de bloco	Gaveta	Propósito
inicializar buracos globais para	Variáveis	Mantenha uma lista de ImageSprites de furos.
criar lista vazia	Listas	Crie uma lista vazia, a ser preenchida quando o programa iniciar.

Comentários (criados clicando com o botão direito em um bloco) são encorajados, mas não obrigatórios.

Iniciando o aplicativo

O primeiro evento a ocorrer em qualquer aplicativo é Screen1.Initialize , portanto, colocaremos o código de inicialização nesse manipulador. Especificamente, adicionaremos os componentes do furo à lista de furos , definiremos a propriedade *Picture* de cada furo como "hole.png" e chamaremos MoveMole . Como ainda não escrevemos o MoveMole, criaremos um procedimento vazio com esse nome, que preencheremos posteriormente.

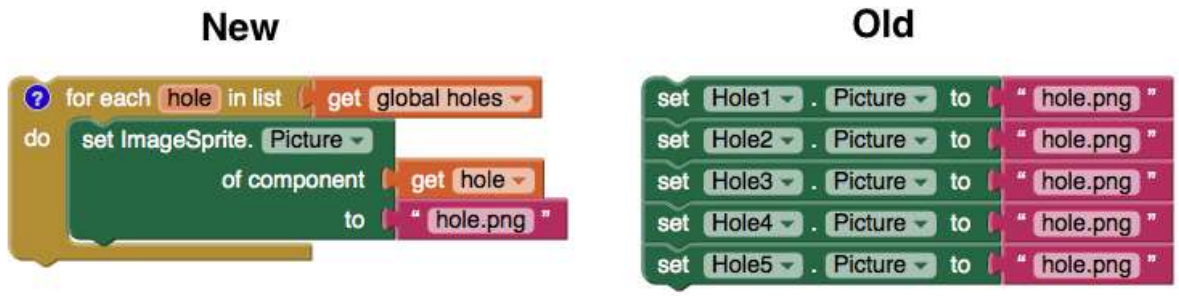
Abaixo está uma tabela dos blocos que você precisa criar. Observe que a gaveta "Any ImageSprite" é encontrada na guia "Any component" na parte inferior da lista de Blocos no Editor de Blocos.



Tipo de bloco	Gaveta	Propósito
Tela1.Inicializar	Tela1	Especifique o que deve acontecer quando o aplicativo for iniciado.

adicionar itens à lista	Listas	Adicione os seguintes valores a...
obter buracos globais	Variáveis	...a lista de buracos
Buraco1	Buraco1	-o buraco superior esquerdo
Buraco2	Buraco2	-o orifício central superior
Buraco3	Buraco3	-o orifício superior direito
Buraco4	Buraco4	-o orifício inferior esquerdo
Buraco5	Buraco5	-o orifício inferior direito
para cada buraco na lista	Ao controle	Percorra a lista de furos .
definir ImageSprite.Imagem do componente ... para	Qualquer ImageSprite	Defina a propriedade Imagem de...
obter buraco global	Variáveis	...o buraco atual ImageSprite
" " (buraco.png)	Texto	...para a imagem do buraco vazio.
para o procedimento (MoveMole)	Procedimentos	Crie um procedimento, a ser preenchido posteriormente, para mover a toupeira.
chamar MoveMole	Procedimentos	Chame o MoveMole para fazer o primeiro posicionamento da toupeira.

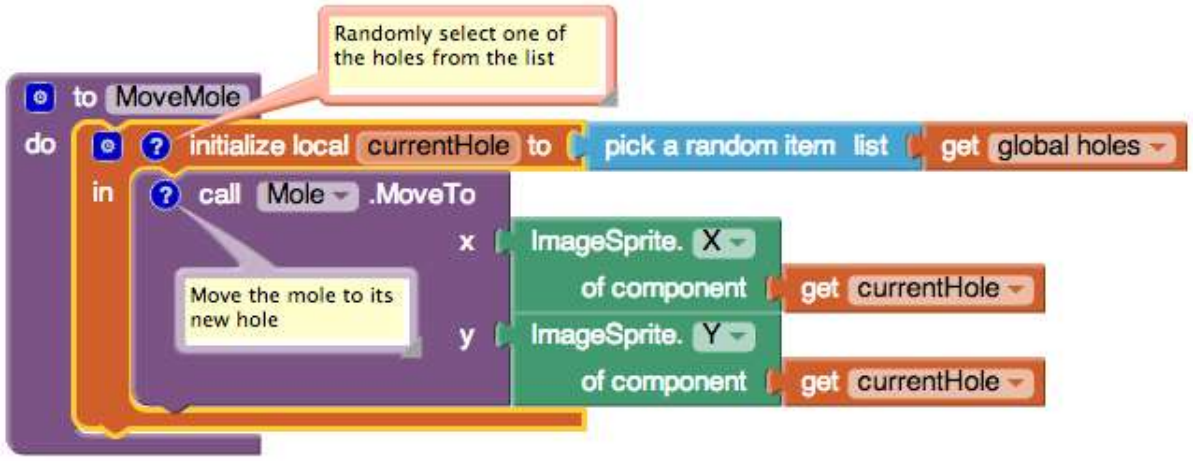
Compare o para cada bloco com os blocos equivalentes que seriam necessários sem ele:



Não só o conjunto esquerdo de blocos é mais curto, como também é menos repetitivo, poupando o programador de copiar e colar semi-estúpido e tornando-o mais fácil de modificar, por exemplo, se o nome da imagem for alterado.

Movendo a toupeira

Agora vamos preencher o corpo do procedimento MoveMole , que chamaremos quando o programa iniciar, quando a toupeira for tocada e quando nosso Timer disparar a cada segundo. O procedimento deve escolher um buraco aleatório e mover a toupeira em cima dele. Aqui estão os blocos compilados e uma tabela dos blocos usados:



Tipo de bloco	Gaveta	Propósito
inicializar local currentHole para (existem dois tipos de 'inicializar local': pegue aquele que se encaixa no bloco de procedimento)	Variáveis	Armazene o valor do furo atual para uso no procedimento.
escolha um item aleatório	Listas	selecione aleatoriamente um buraco da lista
obter buracos globais	Variáveis	lista de buracos ImageSprites
chamar Mole.MoveTo	Verruga	Mova a toupeira para o...
ImageSprite.X do componente	Qualquer ImageSprite	..x-coordenada de...
obter local currentHole	Variáveis	...o buraco escolhido...
ImageSprite.Y do componente	Qualquer ImageSprite	...e a coordenada y.

Agora precisamos especificar que o MoveMole deve ser chamado sempre que o Timer do MoleClock disparar. Só precisamos de dois blocos para conseguir isso:

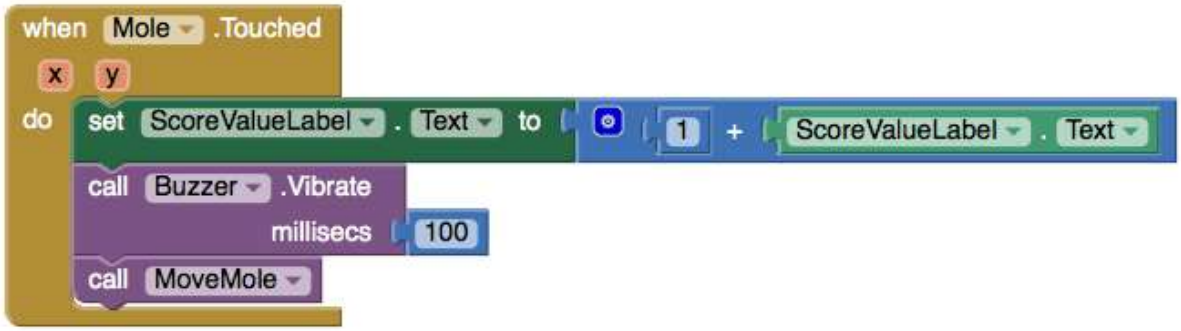
Tipo de bloco	Gaveta	Propósito
quando MoleClock.Timer	MoleClock	Quando o Timer dispara...
chamar MoveMole	Procedimentos	... mova a toupeira.

Registrando Toques

Finalmente, precisamos especificar o que acontece quando a toupeira é tocada. Especificamente, queremos:

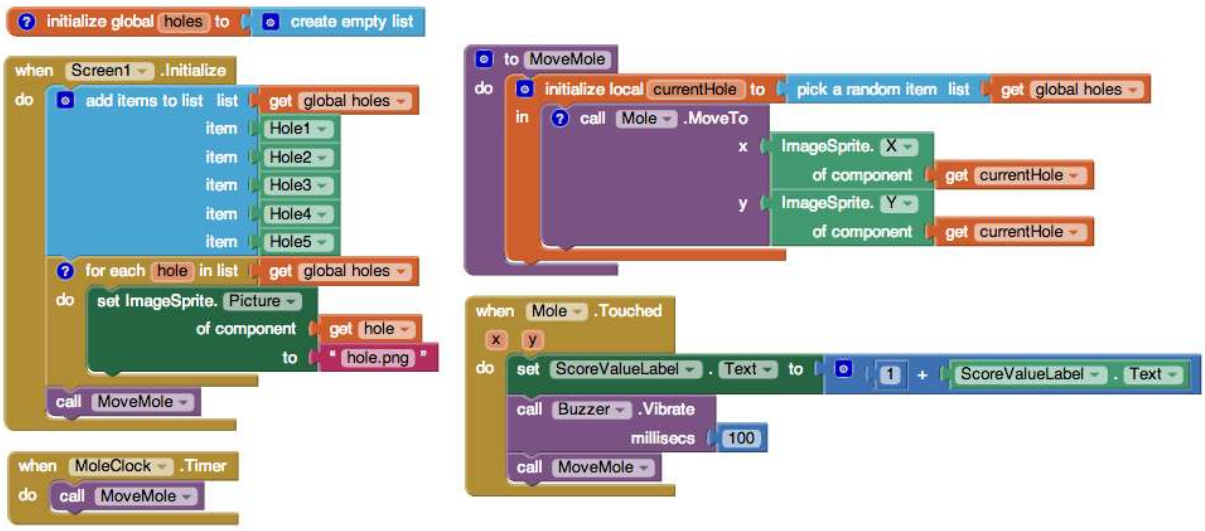
1. Aumente a pontuação.
2. Faça o telefone vibrar brevemente.
3. Mova a toupeira.

Podemos traduzi-los facilmente em blocos:



Tipo de bloco	Gaveta	Propósito
quando toupeira.Tocado	Verruga	Quando a toupeira é tocada...
definir ScoreValueLabel.Text como	ScoreValueLabel	...atualizar a partitura visível para...
+	Matemática	...o resultado da adição...
1	Matemática	...1 [e]...
ScoreValueLabel.Text	ScoreValueLabel	...a pontuação anterior.
chamar Buzzer.Vibrar	campainha	Faça o telefone vibrar por...
100	Matemática	...100 milissegundos.
chamar MoveMole	Procedimentos	Mova a toupeira para um novo local.

Programa final



variações

Aqui estão algumas variações que você pode querer implementar:

- Adicionando um botão Redefinir para definir a pontuação de volta a 0.
- Ter a pontuação depende não apenas do número de acertos, mas também do número de erros e moles escapados.
- Aumentar a velocidade do movimento do mole se o jogador estiver indo bem e diminuindo se o jogador estiver mal.
- Adicionando uma segunda toupeira em um cronômetro diferente.

Você pode ver como implementar as duas primeiras variações no tutorial original do Mole Mash .

Análise

Aqui estão alguns dos conceitos abordados neste tutorial:

- Colocando componentes em uma **lista** .
- Executar uma operação em cada componente em uma **lista** usando os recursos para cada bloco e qualquer componente.
- Colocar um **ImageSprite** em cima do outro, usando sua propriedade *Z* para controlar qual vai na frente.
- Usando o componente **Relógio** para controlar o jogo.
- Criando um procedimento e chamando-o de vários lugares.

`Feito com MoleMash 2 ? Retorne aos outros tutoriais aqui .`

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.

Ou baixe o apk

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.

Suporte do App Inventor: Comunidade
Outras dúvidas: E-mail
GitHub: mit-cml
Acessibilidade: acessibilidade.mit.edu

Texto original

Sugerir uma tradução melhor