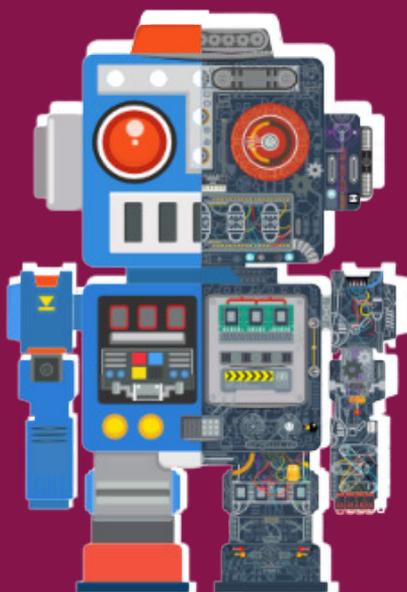


DA ELETRÔNICA À
COMPUTAÇÃO:

Não é mágica, é ciência!

Material extra: tutorial MIT App Inventor



CIÊNCIA E ARTES NAS FÉRIAS

CIÊNCIA E ARTES NAS FÉRIAS

AGRADECIMENTOS

Agradecemos, em especial, a todos os professores e alunos envolvidos nas atividades das oficinas, aos colaboradores e patrocinadores que nos apoiaram e tornaram possível cada detalhe planejado com carinho e atenção, a Pró-Reitoria de Pesquisa (PRP) da Unicamp, pela organização do evento. Por fim, não podemos deixar de agradecer aos alunos que participam das oficinas, afinal, a maior gratificação para nós é vivenciar a alegria de vocês durante e após as atividades.

APOIO



ÍNDICE



Objetivos	_____	04
Antes de começar...	_____	06
App 1 • HelloPurr	_____	10
App 2 • PaintPot	_____	21



OBJETIVOS

O programa "Ciência & Arte nas Férias", promovido pela Unicamp, proporciona durante os meses de férias escolares de verão, um estágio em seus laboratórios para estudantes de escolas públicas de ensino médio da região de Campinas. Todas as grandes áreas do conhecimento estão envolvidas: Artes, Ciências Humanas, Ciências Exatas e da Terra, Ciências Biológicas e da Saúde e Tecnologia. Neste estágio os estudantes se envolvem com os desafios atuais da ciência e da arte, com a metodologia do trabalho científico e da criação artística e com o ambiente humano dos laboratórios de pesquisa.

Na Faculdade de Engenharia Elétrica e de Computação (FEEC) apresentamos uma Oficina de Programação, cujo objetivo é familiarizar os estudantes com alguns conceitos básicos da área. Tentamos aproximar os alunos do nosso ambiente de uma forma mais divertida, para que eles aprendam na oficina como é a lógica básica da programação, mas sem a dificuldade de ter que aprender alguma linguagem específica em pouco tempo, por isso usaremos a plataforma MIT App Inventor – que permite que os alunos desenvolvam os próprios aplicativos android de uma maneira bem simples.

OBJETIVOS

Propomos a construção de aplicativos acompanhados por voluntários do programa, inclusive na finalidade de permitir um contato maior entre os alunos e os atuais estudantes e professores da Unicamp. Esperamos que os alunos aproveitem e consigam despertar a curiosidade que leva a nós, engenheiros, a desenvolver nosso papel na sociedade.

ANTES DE COMEÇAR...

A preparação da programação com o App Inventor é um processo composto de três passos que leva uma meia hora no total.

1. Prepare seu sistema (Java)
2. Instale o Software do App Inventor
3. Inicialize o App Inventor

REQUISITOS DO SISTEMA

Computador e sistema operacional:

Macintosh (com processador Intel): Mac OS X 10.5 ou superior

- Windows: Windows XP, Windows Vista, Windows 7 ou superior
- GNU/Linux: Ubuntu 8+, Debian 5+

Navegador:

- Mozilla Firefox 6.0 ou superior
- Apple Safari 5.0 ou superior
- Google Chrome 4.0 ou superior
- Microsoft internet explorer 7 ou superior

Etapa I: Preparação de seu sistema

O App Inventor requer Java. Verifique se você já tem o programa em seu computador e, para isso, faça a verificação web do Java conforme explicado abaixo. Caso você não possua o programa ainda, siga as instruções de instalação do Java na página da Oracle http://java.com/en/download/help/download_options.xml

Verificação do Java Webstart

Para verificar se o Java Web Start está funcionando corretamente, clique no botão laranja (Launch) que aparece no link abaixo para tentar rodar um programa na Web: <http://appinventor.mit.edu/explore/test-your-system.html>

Este teste deveria baixar da Web e executar um arquivo (notepad.jnlp) que criará uma janela chamada Notepad (ou bloco de notas) na qual você pode introduzir um texto.

Dependendo do seu navegador, talvez você precise abrir manualmente o arquivo quando o download for concluído.

Se o teste der certo, feche o Notepad e continue. Caso contrário, não tente utilizar o App Inventor até o teste dar certo. Tente reinstalar o Java.

Etapa 2: Instale o software do App Inventor

Para poder utilizar o App Inventor, você antes precisa instalar um software em seu computador. Esse software está em um pacote chamado App Inventor Setup.

Siga as instruções para realizar a instalação e, depois, retorne para continuar com os passos 3 e 4.

Instruções para Windows

- Faça o download do instalador no link Download [nesta página](#)
- Localize o arquivo AppInventorSetup_v1_2.exe (aproximadamente 92 MB) na sua pasta de downloads ou área de trabalho.
- Abra o arquivo de instalação.
- Siga os passos da instalação. Não modifique o local da instalação, apenas guarde o diretório, pois, mais tarde, você deve checar o driver. O diretório de instalação pode ser diferente conforme sua versão do Windows e se você está logado como administrador ou não.

Etapa 3: Inicialização do App Inventor

Inicie o software no pacote App Inventor Setup que foi baixado no passo 2. Tenha certeza de que você tem acesso à internet e que você tenha uma conta do Gmail para continuar.

Inicialize o designer e crie um novo projeto

Em seu navegador, abra o site do App Inventor.

Se essa é a primeira vez que utiliza o App Inventor, a página de projetos estará em branco

- Clique em New na parte superior esquerda da página
- Entre com o nome do projeto na caixa de diálogo que aparece e clique em OK.

O navegador abrirá um site chamado Designer no qual você deve selecionar os componentes para o app e desenhar a interface de usuário. Agora você já pode seguir para os tutoriais de aplicativos!

Observação: a instalação não deu certo ou seu sistema operacional/celular não foi coberto aqui? Acesse o manual completo em Português no link:

<http://www.colmenia.org/documents/626016/632167/Guia+de+iniciaci%C3%B3n+e+instalaci%C3%B3n+de+App+Inventor+pb+re+vision+5.pdf/40c9005e-bc57-418f-8258-dad1823b7c2d>

Esta apostila é uma tradução livre feita pelos voluntários do CAF de tutoriais disponibilizados pelo MIT no site. Neste site podem ser encontrados ainda mais tutoriais em inglês.

Quer aprender mais sobre programação? Confira o site e escolha o que aprender a seguir!

App I • HelloPurr

Agora que você já preparou o seu computador e o seu dispositivo e aprendeu como o Designer e o Block Editor funcionam, você está pronto para desenvolver o seu app HelloPurr. A esta altura você já deve ter o Designer ou o Block Editor aberto no seu navegador, além de um dispositivo ou emulador Android conectado ao Block Editor (veja instruções de instalação para App Inventor 2 se você não tem isto rodando).

HelloPurr: Clique no gatinho e ouça ele miar

O HelloPurr é um app simples que você pode desenvolver em um curto espaço de tempo. Você cria um botão com a imagem de um gato e programa este botão para tocar o som de “miau” ao ser clicado.

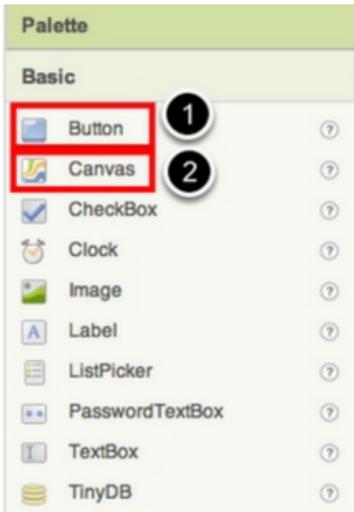
Para desenvolver o HelloPurr, você vai precisar de um arquivo de imagem de um gato e um arquivo de áudio com o som de “miau”. Clique nos links a seguir e faça o download destes arquivos para o seu computador. Para fazer o download: após clicar no link, clique com o botão direito na imagem ou na barra de áudio e selecione “Salvar como”. Salve ambos os arquivos na sua área de trabalho ou na sua pasta de downloads.

- Imagem gatinho: [gatinho.png](#) (clique com o botão direito e salve)
- Áudio miau: [miau.mp3](#) (clique com o botão direito e salve)

Selecione os componentes para desenhar seu app

Os componentes do App Inventor estão localizados no lado esquerdo do Designer Window sob o título Palette.

App I • HelloPurr



Os componentes são os elementos básicos usados para desenvolver apps em dispositivos Androids. Eles são como os ingredientes em uma receita. Alguns componentes são bem simples, como o componente Label, que apenas mostra o texto na tela, ou o componente Button (#1 à esquerda) que você clica para iniciar uma ação.

Outros componentes são mais elaborados: a tela Canvas onde você pode abrigar imagens ou animações, o sensor Accelerometer que funciona como um controlador Wii e detecta movimentos no aparelho, componentes que enviam mensagens de texto, componentes que reproduzem música e vídeo, componentes que pegam informação de web sites, e por aí vai.

Para usar um componente no seu app, você precisa clicar sobre ele e arrastá-lo para a tela de visualização localizada no Designer. Quando você adiciona um componente para o Viewer (#1 abaixo), ele também aparece na lista de componentes a direita do Viewer.

Os componentes (#2 abaixo) tem propriedades que podem ser ajustadas para mudar a sua aparência e o seu comportamento no app. Para visualizar e alterar as propriedades de um componente (#3 abaixo), primeiro você seleciona o componente na lista.

App I • HelloPurr



Passo a passo na seleção dos componentes e suas propriedades

O HelloPurr terá um componente Button que vai exibir a imagem do gato que foi baixado anteriormente. Para isso:

1. Da paleta User Interface, arraste e solte o componente Button para Screen1 (#1).

2. Para fazer com que o botão tenha a imagem de um gato, vá no painel Properties e em seguida em Image e clique no texto "None..." e em seguida em "Upload New..." (#2). Vai abrir uma janela para você escolher o arquivo da imagem. Clique em "Procurar" e então navegue até o local onde foi o salvo o arquivo gatinho.png. Clique no arquivo e em seguida em "Abrir" e "OK".



App I • HelloPurr

3. Mude o texto do componente Button: apague "Text for Button1" e deixe o espaço do texto em branco, assim não haverá nenhum texto sobre a imagem do gatinho. O seu Designer deve aparecer dessa forma:



Se a imagem do gatinho aparecer cortada, você pode consertá-la nas propriedades de Height e Width no botão de "Fill Parent". Para fazer isso, clique no componente Button, vá até o painel Properties à direita da tela, desça a barra de rolagem até a propriedade Width e em seguida clique em "Automatic..." para ativar a lista. Escolha a opção "Fill Parent". Faça o mesmo para a propriedade Height.

4. Na paleta User Interface, arraste e solte o componente Label no Viewer (#1) embaixo da imagem do gatinho. Ele vai aparecer na sua lista de componentes como Label1.

Você vai ver o texto mudar no Designer e no seu dispositivo. Mude o FontSize do Label1 para 30 (#3). Clique na caixa (#4) e mude o BackgroundColor do Label1: você pode alterar a cor de fundo para qualquer outra cor. Mude o TextColor do Label1 (#5) para a cor desejada. Aqui, a cor de fundo está em azul e a cor do texto em amarelo.

App I • HelloPurr



5. Em Palette, clique na aba Media e arraste o componente Sound para o Viewer (#1). Independentemente da onde você soltar o componente, ele vai aparecer na área embaixo do Viewer marcado como Non-visible components. No painel Media, clique em Upload New... (#2), procure pelo local onde foi salvo o arquivo miau.mp3 e faça o upload para este projeto (#3). No painel Properties, observe que na propriedade Source está escrito "None...". Clique na palavra "None..." e altere o componente Source do Sound1 para miau.mp3 (#4).



App I • HelloPurr

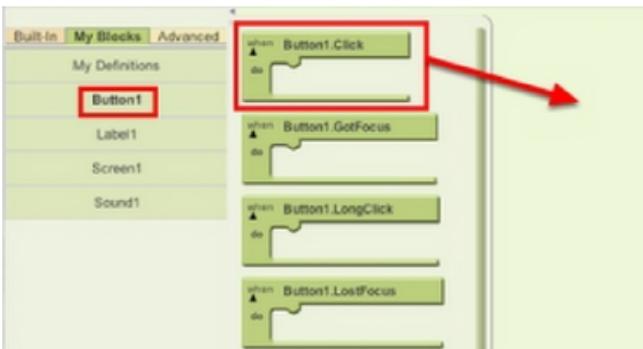
Programando com o Blocks Editor

Até agora você organizou a tela e os componentes do seu app no Designer, localizado no navegador. Para começar a programar o seu app você vai precisar do Blocks Editor. Clique no botão Blocks para ir até o Blocks Editor.

Uma vez com o Blocks Editor aberto, siga para o próximo passo para começar a programar o seu app com os blocos.

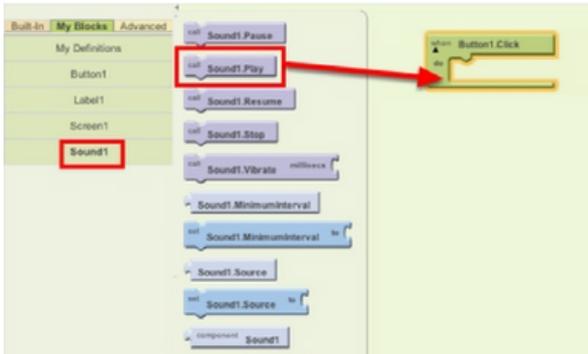
Fazendo o som tocar

1. No lado esquerdo do Blocks Editor, clique na aba Button1 para abri-la. Arraste e solte o bloco Button1.Click na área de trabalho (a área aberta à direita). Os blocos de cor amarelo mostarda são chamados de blocos event handler. Estes blocos especificam como o dispositivo deve responder a certos eventos: um botão é pressionado, o dispositivo é sacudido, o usuário arrasta o dedo sobre uma imagem, etc. Os blocos event handler são amarelo mostarda e usam a palavra when. Por exemplo, when Button1.Click é um event handler.



App 1 • HelloPurr

2. Clique na aba Sound1, arraste o bloco Sound1.Play e o conecte à seção “do” do bloco Button1.Click. Os blocos se conectam entre si como peças de um quebra-cabeça e você vai ouvir um som quando eles se encaixarem.



Os blocos de cor roxa são chamados blocos de comando, que são encaixados no corpo dos blocos event handler. Quando um event handler é executado, ele segue a sequência de comandos especificada no corpo do bloco. Um comando é um bloco que especifica uma ação a ser realizada (por exemplo, tocar um som) quando um evento (por exemplo, pressionar Button1) é acionado.

A este ponto, os seus blocos devem se apresentar dessa forma:



Agora você pode observar que o bloco comando está dentro do event handler. Este conjunto de blocos significa: “quando Button1 for clicado, Sound1 vai tocar”. O event handler é como uma categoria de ação (por exemplo, um botão é clicado), e o comando especifica o tipo e os detalhes da ação (por exemplo, tocar som, som especificado).

App I • HelloPurr

Experimente! Quando você clicar no botão você deverá ouvir o gatinho miar. Parabéns, o seu primeiro app está rodando!

Nota: existe um problema conhecido com o componente Sound em alguns dispositivos. Se você receber uma mensagem “OS Error” e o som não tocar – ou se houver um atraso no som, volte ao Designer e tente usar o componente Player (encontrado em Media) ao invés do componente Sound.

Salvando seu aplicativo

Enquanto o seu dispositivo (emulador ou telefone/tablete) estiver conectado com o App Inventor, o seu app está rodando em tempo real. Se você desconectar o emulador/telefone/tablete do Blocks Editor, o app vai desaparecer. Você pode trazê-lo de volta reconectando o dispositivo. Para manter o app rodando sem estar conectado ao App Inventor, você precisa “package” o app para produzir um application package (arquivo apk).

Para “package” o app ao seu telefone ou enviá-lo para outra pessoa, clique na aba Build no topo da tela.

Em Build existem duas opções disponíveis para você escolher:

1. App (provide QR code): Você pode gerar um código de barras (código QR), que pode ser usado para instalar o app em um telefone ou tablete que possua uma câmera, com a opção de escaner de código QR, como o ZXing barcode scanner (disponível gratuitamente no Google Play).

App I • HelloPurr

Nota: Este código de barras funciona somente para o seu dispositivo, pois ele está associado à sua conta do google. Se você deseja compartilhar com terceiros através do código de barras, você vai precisar fazer o download do arquivo .apk para o seu computador e utilizar um software para converter o arquivo em um código de barras.

2. App (save to my computer): Você pode fazer o download do app para o seu computador como um arquivo apk. Você pode distribuí-lo e compartilhá-lo como quiser, instalando manualmente o app em outros dispositivos (também chamado de “side loading”).

Desafio: Faça o gato ronronar!

O desafio é fazer o gato ronronar ao clicar o botão. Vá até o Blocks Editor, abra a aba Sound1, arraste e solte o bloco Sound1.Vibrate abaixo do bloco Sound1.Play.



O bloco Sound1.Vibrate possui um encaixe em aberto, o que significa que você precisa encaixar outra peça que especifique como o bloco vai funcionar. Neste caso, queremos especificar a duração da vibração. Números são calculados em milésimos de segundo: para fazer o telefone vibrar por meio segundo é preciso encaixar uma peça no valor de 500 milissegundos.

App I • HelloPurr

Na paleta Built-In, vá até a aba Math, arraste o bloco número e posicione no encaixe do bloco Sound1.Vibrate.



Após posicionar o bloco número, clique no número "0". Com o número grifado, digite "500" com o teclado.



Pronto!

Agora conecte o seu telefone e clique na imagem do gato na tela. O telefone deverá vibrar e fazer miau ao mesmo tempo. Se você salvou o seu app ao seu telefone, isto pode não funcionar. Apesar de parecer que o seu telefone está conectado, ele pode não estar conectado em tempo real. Para reconectar o seu telefone, clique no botão Connect no topo da tela e selecione Reset Connection. Se não funcionar, saia do app no seu telefone e reinicie a conexão.

Aqui estão as principais ideias abordadas até agora:

- Você desenvolve apps através da seleção de componentes (ingredientes) e a partir do que e quando você manda ele executar.
- Você usa o Designer para selecionar componentes e estabelece as suas propriedades. Alguns componentes são visíveis e outros não.
- Você pode fazer o upload de sons e imagens e adicioná-los aos apps.
- Você usa o Blocks Editor para construir os blocos e programar os componentes.
- Blocos when... do... definem event handlers (gerenciadores de eventos), que ditam aos componentes o que fazer quando algo acontecer.
- Blocos call... estabelecem uma ação aos componentes.

App 2 • PaintPot

Este tutorial introduz a camada Canvas para criar gráficos simples de duas dimensões. Você construirá um aplicativo que permitirá desenhar na tela do celular em cores diferentes.

Com PaintPot você pode:

- Mergulhar o dedo em um balde de tinta virtual para desenhar naquela cor
- Arrastar seu dedo pela tela para criar uma linha
- Tocar na tela para criar pontos
- Usar o botão na parte inferior para limpar a tela toda
- Incluir uma imagem como plano de fundo para desenho

Este tutorial assume que você completou o tutorial HelloPurr. Esse tutorial introduz os seguintes conceitos do App Inventor:

- A camada “canvas” para desenhar
- Controlar o layout da tela com os componentes “Arrangement”
- Gerenciadores de eventos que exigem argumentos
- Variáveis



Antes de começar:

Verifique se seu computador e seu telefone estão configurados para usar o App Inventor. Comece um novo projeto na janela Designer e chame-o de “PaintPot”. Abra o Editor de Blocos, clique em Connect to Phone, e verifique se o telefone iniciou o app do App Inventor.

App 2 • PaintPot

Título da tela

Para começar, vá ao painel “Properties” à direita do “Designer” e mude o título da tela para “PaintPot”. Você deve ver essa mudança no telefone, com um novo título mostrado na barra de títulos.

Existem três nomes no App Inventor e é fácil confundí-los:

1- O nome que você escolhe para o projeto enquanto você trabalha nele (nesse caso, PainPot). Esse também será o nome da aplicação se você salvá-la pala o telefone.

2- O nome “Screen1”, que é o nome da componente Canvas. Você verá ele listado no painel Components em Designer. Você não pode mudar o nome da primeira componente Screen na versão atual do App Inventor, mas você pode criar telas adicionais e nomeá-las como você quiser.

3- A propriedade Title da tela, que é o que você vai ver na pequena barra de título do telefone. Título é uma propriedade da componente Screen. O Title começa sendo “Screen1”, que é o que você usou no HelloPurr. Entretanto, você pode mudá-lo, como você está fazendo no PaintPot. Para reiterar, o nome e o título de Screen1 são inicialmente os mesmos, mas você pode mudar o título se você quiser.

Configurando os componentes

Você utilizará esses componentes para criar o PaintPot:

- Três botões para selecionar tinta vermelha, azul ou verde (red, green, blue) e outro botão para limpar o desenho (botão wipe)

App 2 • PaintPot

- Um Canvas, a superfície de desenho. Essa tela tem uma imagem de fundo (BackgroundImage), que é o gatinho do tutorial do HelloPurr. Você pode desenhar numa tela preta também. Essa é só uma tela sem uma imagem de fundo.

Tem um componente que você não vê: você usa um 'HorizontalArrangement' para fazer os 3 botões das cores ficarem alinhados.

Então são 5 componentes no total. Vamos pegá-los e construir o app.

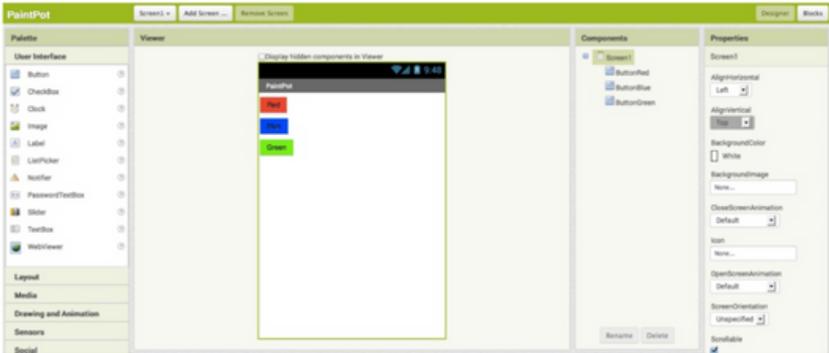
Botões coloridos

- Arraste um componente "Button" até o Viewer e mude o atributo texto do botão para "Vermelho" e faça sua cor de fundo vermelha.
- Clique em "Button1" na lista de componentes no Viewer para destacá-lo e use o botão "Rename..." para mudar o seu nome de "Button1" para "RedButton"
- Similarmente, faça mais dois botões para verde e azul, chamados "BlueButton" e "GreenButton", colocando-os verticalmente embaixo do botão vermelho.

A seguir, um print de como ele deve ficar no Designer, com os nomes dos botões aparecendo na lista dos componentes do projeto. Neste projeto, você está mudando os nomes dos componentes em vez de deixar os nomes padrão como você fez no HelloPurr. Usar nomes com significado faz seus projetos mais legíveis para você e para os outros.

Você também deve ver os 3 botões na tela do celular.

App 2 • PaintPot



Layout com Screen Arrangement

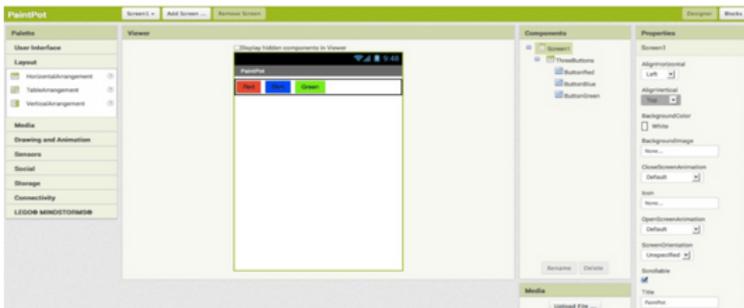
Agora você deve ter três botões, um acima do outro. O próximo passo é deixá-los alinhados horizontalmente. Faz-se isso usando um componente HorizontalArrangement.

1. Da categoria , puxe HorizontalArrangement e coloque em baixo dos botões. Mude o nome desse componente de "HorizontalArrangement1" para "ThreeButtons"
 2. No painel de Properties, mude a largura (width) de "ThreeButtons" para "Fill Parent...", para que ele ocupe toda a largura da tela
 3. Mova os três botões lado a lado no componente HorizontalArrangement. Você verá uma linha vertical azul mostrando para onde irá a peça que você está arrastando.
- Se você olhar na lista de componentes do projeto, você verá que os três botões estão indentados para mostrar que eles agora são subcomponentes. Note que todos os componentes estão indentados abaixo de Screen1.

App 2 • PaintPot

Você deve também ver seus três botões alinhados em uma linha na tela do telefone, mas talvez as coisas fiquem um pouco diferentes do que elas ficam no Designer. Por exemplo, o contorno do Arrangement é mostrado no Designer mas não no telefone.

Em geral, você usa o Screen Arrangement para criar layouts verticais ou horizontais simples. Você pode criar layout mais complexos encaixando componentes de Screen Arrangement. Existe também um componente chamado TableArrangement, não incluso neste tutorial.



Canvas e botão Wipe

Os dois componentes finais são o Canvas e o botão Wipe.

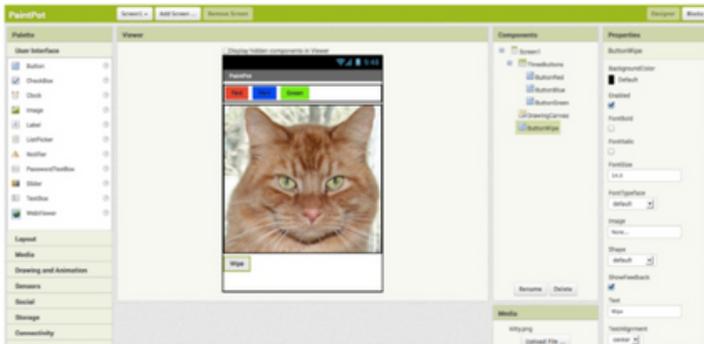
1. A partir de arraste um componente Canvas para o Viewer. Mude seu nome para "DrawingCanvas". Defina sua largura (width) para "Fill Parent..." e defina sua altura (Height) para 300 pixels.
2. Adicione uma imagem de fundo ao Canvas. Clique no campo contendo "None..." perto de BackgroundImage no painel de propriedades da tela. Você pode usar o mesmo arquivo kitty.png do outro tutorial ou outra imagem.

App 2 • PaintPot

Dica: você pode usar qualquer imagem que quiser, mas você terá resultados melhores se o tamanho da imagem (em pixels) for próximo ao tamanho da imagem que você verá no telefone. Imagens muito grandes demoram muito para carregar e podem exceder a capacidade de memória que o celular aloca para aplicativos.

3. De Pallette, arraste o botão final para a tela, colocando-o embaixo da tela. Renomeie esse componente de botão para “buttonwipe” e mude seu atributo texto para “Wipe”

Você agora completou os passos para configurar a aparência de seu aplicativo. Aqui é como ele deve ficar no Designer. A seguir, você definirá como os componentes se comportam.



Adicionar Comportamentos aos Componentes

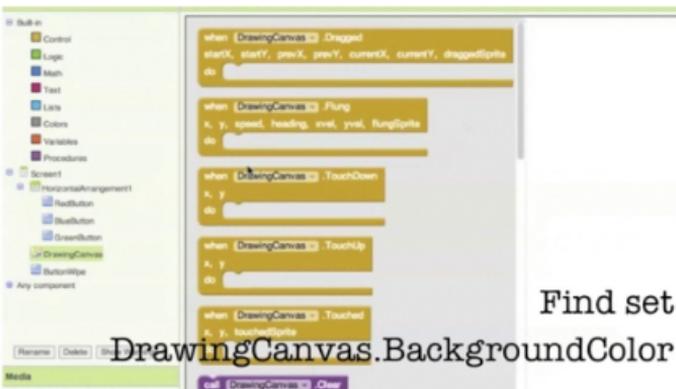
Clique no botão Blocks para mudar para o editor de blocos. Primeiro você vai definir os botões que mudam a cor da tinta. Depois você irá adicionar blocos para decidir o que acontece quando alguém toca ou arrasta o dedo sobre a tela.

App 2 • PaintPot

Adicionar botões gerenciadores de eventos

No editor de blocos:

1. Abra a pasta RedButton e arraste o bloco “when ButtonRed.click”
2. Abra a pasta DrawingCanvas. Arraste o bloco “set Drawingcanvas.PaintColor to” (lembre que o conjunto de componentes de bloco é desdobrável e PaintColor é uma seleção nesse menu) e coloque-o na seção “do” de “when ButtonRed.click”.



App 2 • PaintPot

3. Abra a pasta de cores e arraste pra fora e coloque em “set DrawingCanvas.PaintColor to” (Clicar em um bloco de cor depois de ter sido colocado irá exibir uma tabela de cores que você pode selecionar.)

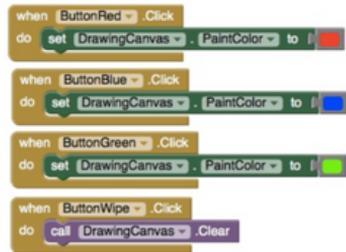
4. Repita os passos 2-4 para os botões azul e verde.

5. O botão final a ser configurado é o botão Limpar. Faça um gerenciador de eventos para ButtonWipe arrastando “when ButtonWipe.click” da pasta ButtonWipe. Da pasta DrawingCanvas arraste “call DrawingCanvas.clear” e coloque-o na área “do” do bloco “when ButtonWipe.click”

Os blocos devem ficar assim:

Adicionar gerenciadores de eventos de toque

No editor de Blocos, abra a pasta da tela e arraste o bloco “when DrawingCanvas.Touched” para a área de trabalho.



Assim que você arrastar o bloco, você perceberá 3 nomes de argumentos (destacados em laranja) localizados no topo do bloco: x,y e touchedSprite. Esses argumentos são também conhecidos como variáveis locais e podem ser acessados usando os blocos get e set encontrados na pasta de Variáveis e então selecionando a variável apropriada no menu desdobrável. Você também pode acessar essas variáveis movendo o cursor para as áreas de nome destacado e selecionando qual a variável que você gostaria de usar.

App 2 • PaintPot

Você já viu eventos de clique de botão. Cliques são simples, porque não há nada para saber sobre o clique além do que quando ele aconteceu. Outros gerenciadores de eventos como “when...touched” precisam da informação sobre o evento. No App Inventor, essa informação é expressa como o valor dos argumentos associados com o gerenciador de eventos. Para o evento “when...touched”, os dois primeiros argumentos são as coordenadas x e y de onde ocorre o toque. Guardaremos o TouchedSprite para outro tutorial.

Para este evento de toque, faça a tela desenhar um pequeno círculo com as coordenadas (x,y) . Arraste um comando “call DrawingCanvas.DrawCircle” da pasta da tela e coloque-o na seção “do” de “when DrawingCanvas.touched” Para este evento de toque, faça a tela desenhar um pequeno círculo com as coordenadas (x,y) . Arraste um comando “call DrawingCanvas.DrawCircle” da pasta da tela e coloque-o na seção “do” de “when DrawingCanvas.touched”

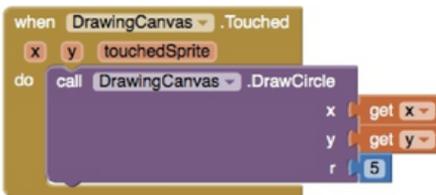
Do lado direito de “call DrawingCanvas.DrawCircle” têm 3 espaços onde você deve especificar os valores de x e y onde o círculo deve ser desenhado e r , o raio do círculo. Para x e y você usará os valores dos argumentos que são fornecidos pelo gerenciador de eventos de Toque.

1. Mova seu cursor até a variável x (destacada em laranja). Encontre o bloco “get x ” e arraste-o até o espaço de x correspondente no bloco “when DrawingCanvas.touched”.
2. Faça o mesmo para a variável y .
3. Você também precisa definir o raio do círculo que será desenhado. 5 (pixels) é um bom valor pra esse app. Clique na área em branco da tela e digite o número 5, que criará um

App 2 • PaintPot

bloco com o valor 5. Digitar na área em branco é chamado “typeblocking” e é um atalho útil que pode ser usado. Isso pode ser feito para qualquer bloco, não apenas números. Agora encaixe o bloco com valor 5 no espaço do raio.

Aqui é como o gerenciador de eventos de toque deve ficar:



Teste o que você tem até agora no telefone. Aperte um botão de cor. Agora toque a tela. Seu dedo deve deixar um ponto no lugar onde ele tocou.

Apertar o botão “wipe” deve limpar o desenho que você fez.

Adicionar eventos de arrastar

Finalmente, o gerenciador de eventos de arrastar. Essa é a diferença entre arrastar e tocar:

- um toque é quando você coloca seu dedo na tela e levanta-o sem movê-lo
- arrastar é quando você coloca seu dedo na tela e move o dedo enquanto ainda faz contato

Quando você arrasta seu dedo na tela parece que você desenha uma curva enorme onde o seu dedo tocou. O que acontece na verdade é que você está desenhando várias pequenas retas: cada vez que você move um pouco seu dedo, você estende a linha desde a última posição do seu dedo para a posição atual.

App 2 • PaintPot

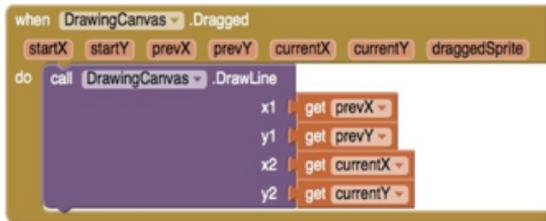
Um evento de arrastar tem 6 argumentos. São pares de coordenadas x e y que mostram:

- a posição (x,y) do seu dedo onde você começou a arrastar
- a posição atual (x,y) do seu dedo
- a posição (x,y) imediatamente anterior do seu dedo

Agora faça o ato de arrastar criar uma linha entre a posição anterior e a nova por meio de um gerenciador de eventos:

1. Da pasta DrawingCanvas, arraste o bloco “when DrawingCanvas.Dragged” para a área de trabalho
2. Ainda da pasta DrawingCanvas, arraste o bloco “call DrawingCanvas.drawline” para o espaço “do” do bloco “when DrawingCanvas.Dragged”
3. Arraste o bloco “get prevX” para o espaço x1 em “call DrawingCanvas.drawline”. Faça o mesmo para os outros espaços: y1 deve ser “get prevY” e x2 deve ser “get currentX” e y2 deve ser “get currentY”.

Aqui o resultado:



Teste seu trabalho no telefone: arraste seu dedo pela tela e desenhe linhas e curvas. Toque a tela para fazer pontos. Use o botão wipe para limpar a tela.

App 2 • PaintPot

Criando variáveis

O tamanho dos pontos desenhados na tela é determinado pelo gerenciador de eventos `when DrawingCanvas.Touched` onde `call Drawing.DrawCircle` utiliza o valor de `r`, que é o raio do círculo, igual a 5. Para mudar a espessura do traço, tudo o que precisamos fazer é utilizar diferentes valores para `r`. Use `r = 2` para pequenos pontos e `r = 8` para pontos maiores.

Comece criando nomes para estes valores:

1. Abra o Editor de Blocos (Blocks Editor) se ainda não estiver aberto e conecte o celular. O seu celular deve mostrar a tela e os botões que você tinha colocado anteriormente.

2. No Editor de Blocos (Blocks Editor), na coluna Built-In, abra o Criador de Variáveis (Variables Drawer). Arraste um bloco inicialize global name to para a tela. Mude o texto "name" para "small". Um sinal de exclamação amarelo pode aparecer no bloco. Se você colocar o mouse em cima, você verá uma mensagem de advertência avisando que o bloco tem um canto vazio.

3. Você precise preencher esse canto vazio com um bloco contendo um número, que especifica o valor para "small" – use 2 como valor. Para criar um bloco numérico, apenas digite o número 2. Um menu vai aparecer, mostrando todos os blocos possíveis que tem um "2" no nome. Clique no primeiro, que é apenas o número 2, e o bloco deve aparecer na tela. Conecte com o bloco criado anteriormente. A marcação amarela de advertência desaparecerá, pois o canto não estará mais vazio (o Segundo valor listado no menu é o bloco matemático `atan2` que não usaremos aqui)

App 2 • PaintPot

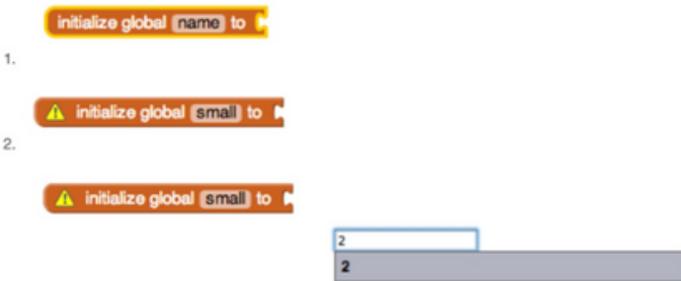
Estes são os passos em sequência:

O tamanho dos pontos desenhados na tela é determinado pelo gerenciador de eventos `when DrawingCanvas.Touched` onde `call Drawing.DrawCircle` utiliza o valor de `r`, que é o raio do círculo, igual a 5. Para mudar a espessura do traço, tudo o que precisamos fazer é utilizar diferentes valores para `r`. Use `r = 2` para pequenos pontos e `r = 8` para pontos maiores.

Comece criando nomes para estes valores:

1. Abra o Editor de Blocos (Blocks Editor) se ainda não estiver aberto e conecte o celular. O seu celular deve mostrar a tela e os botões que você tinha colocado anteriormente.

2. No Editor de Blocos (Blocks Editor), na coluna Built-In, abra o Criador de Variáveis (Variables Drawer). Arraste um bloco `initialize global name to` para a tela. Mude o texto "name" para "small". Um sinal de exclamação amarelo pode aparecer no bloco. Se você colocar o mouse em cima, você verá uma mensagem de advertência avisando que o bloco tem um canto vazio.



App 2 • PaintPot



Você acaba de definir uma variável global chamada `small` cujo valor é o número 2. Semelhante ao `small`, defina uma variável global chamada `big` e dê a ela o valor 8. Finalmente, defina uma variável global chamada `dotsize` e dê a ela o valor inicial de 2.

Você pode se perguntar se não seria um melhor estilo de programação para fazer o valor inicial de `dotsize` ser o valor de `small`, em vez de 2. Isso seria verdade, com exceção de um ponto programação sutil: Fazer isso seria contando com a suposição de que `small` vai já tem um valor em tempo quando `dotsize` recebe seu valor. No App Inventor, você não pode fazer suposições sobre a ordem em que diferentes blocos def serão processados. Em geral, é claro, que você realmente gostaria de especificar a ordem em que as variáveis são atribuídas. Você pode fazer isso através da atribuição de todos os valores assim que o aplicativo é inicializado, usando o evento "Screen initialize". O Quiz Me tutorial dá um exemplo de inicialização.



App 2 • PaintPot

Usando variáveis

Agora volte ao “touch event handler” que você criou na Parte 1 e modifique o bloco call to DrawCircle para que ele use o valor de dotsize ao invés de usar sempre o valor 5.

No editor de blocos (Blocks Editor), abra o criador de variáveis (Variables drawer). Puxe um bloco get e clique no menu suspenso. Você deve ver três novas variáveis no menu suspenso: small, big e dotsize.

Esses blocos foram automaticamente criados e colocados nos menus dos blocos de variáveis de get e set, de modo semelhante que x e y foram criados no menu quando você definiu o gerenciador de eventos (event handler) when DrawingCanvas.Touched na primeira parte desse tutorial. “Global” significa “variável global”, o que é diferente dos argumentos dos event-handler, que são considerados “variáveis locais”. A diferença é que os valores de argumento somente são acessíveis dentro do gerenciador daquele evento, enquanto as variáveis globais são acessíveis por todo o programa.

- Vá para o when MyCanvas.Touched e substitua o bloco de número 5 no call DrawCircle pelo bloco get dotsize .

Mudando o valor de variáveis

Agora crie um modo de trocar o tamanho dos pontos (dotsize) para small ou big. Faça isso com botões.

1. No Designer, arraste um componente Horizontal Arrangement para a tela, abaixo do DrawingCanvas. Dê o nome “BottomButtons” para o componente.

App 2 • PaintPot

2. Arraste o ButtonWipe (já existente) para o BottomButtons.
3. Arraste mais dois botões da paleta para o BottomButtons, colocando-os perto do ButtonWipe
4. Nomeie os botões para “ButtonBig” e “ButtonSmall” e coloque o texto de exibição deles para “Big dots” e “Small dots”, respectivamente.
5. No Editor de Blocos (Blocks Editor), crie um evento when... Clicked para o ButtonSmall que muda o valor do dotsize para o valor de small. Para modificar dotsize, use o bloco set global dotsize to que está no menu MyDefinitions e conecte no bloco global small.
6. Crie um evento similar para ButtonBig.

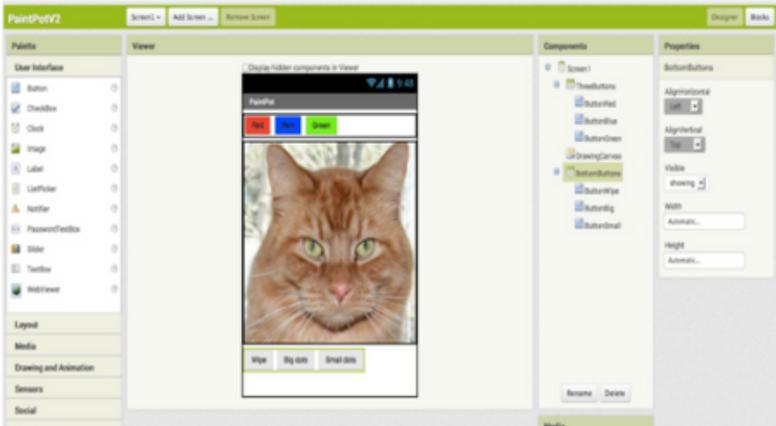
Os dois eventos devem ficar semelhantes a:



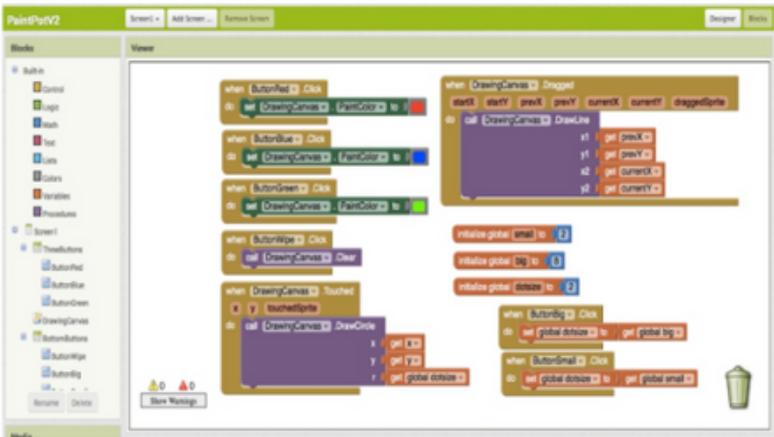
Você terminou! Agora você pode desenhar no PaintPot e utilizar os novos botões para fazer pontos pequenos ou grandes. Perceba que arrastar seu dedo ainda produz uma linha fina. Isso acontece porque as mudanças que fizemos aqui não influenciam o DrawLine.

Aqui está o programa finalizado no Designer:

App 2 • PaintPot



Em no editor de blocos:



Um bug: O programa que você acabou de criar tem um pequeno bug. Se você começar a desenhar antes de pressionar algum dos botões de desenho, a cor selecionada será preto. Entretanto, após escolher alguma outra cor, não tem como voltar para o preto. Pense sobre como você poderia arrumar este problema.



Você cria variáveis globais usando o bloco def no Seletor de Variáveis (Variables Drawer).

Para cada variável global que você definir, o App Inventor automaticamente fornece um bloco global que insere um valor na variável, e um bloco set global ... to para modificar o valor dela. Esses blocos podem todos ser encontrados no Seletor de Variáveis (Variables Drawer).