



Take a picture with Tim!

An MIT App Inventor tutorial

Feat. Tim the beaver



App overview: Take a picture with Tim!

When you are done you and your friends will be able to use this app to:

- Take a picture of yourselves using App Inventor's camera component
 - Set this picture as the background of a *Canvas* component
- Add an image of Tim the Beaver and place him wherever you would like in the picture!

Let's get started!

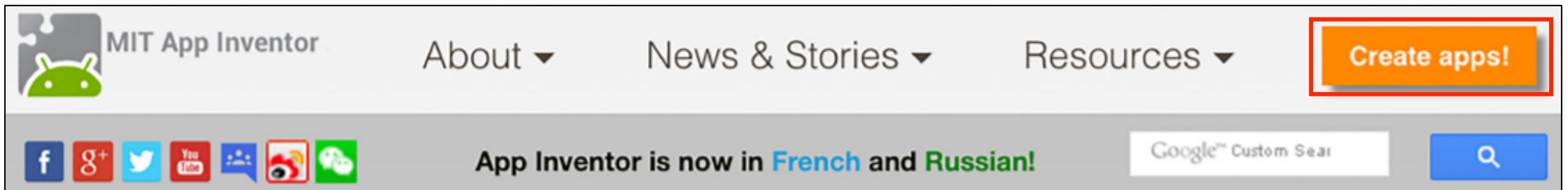




MIT App Inventor

Step 1: Signing in to App Inventor

Click the “Create apps!” button in the menu bar at the top of the MIT App Inventor Hour of Code page.



Step 1 continued

Welcome to MIT App Inventor!

You can either Continue with an Account, and you will be given a Revisit Code to return to the site if you wish.



Continue Without An Account

or

Your Revisit Code:

 - - -

Enter with Revisit Code

Or you can sign in if you have a Google account. Your projects will be saved with your account id.



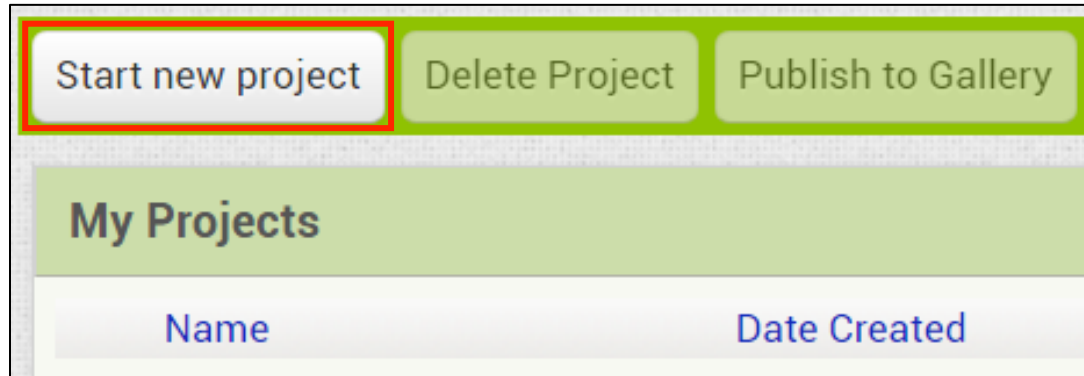
Or sign in with:





Step 2: Creating a new project

Click “Start a new project” in the upper left corner...



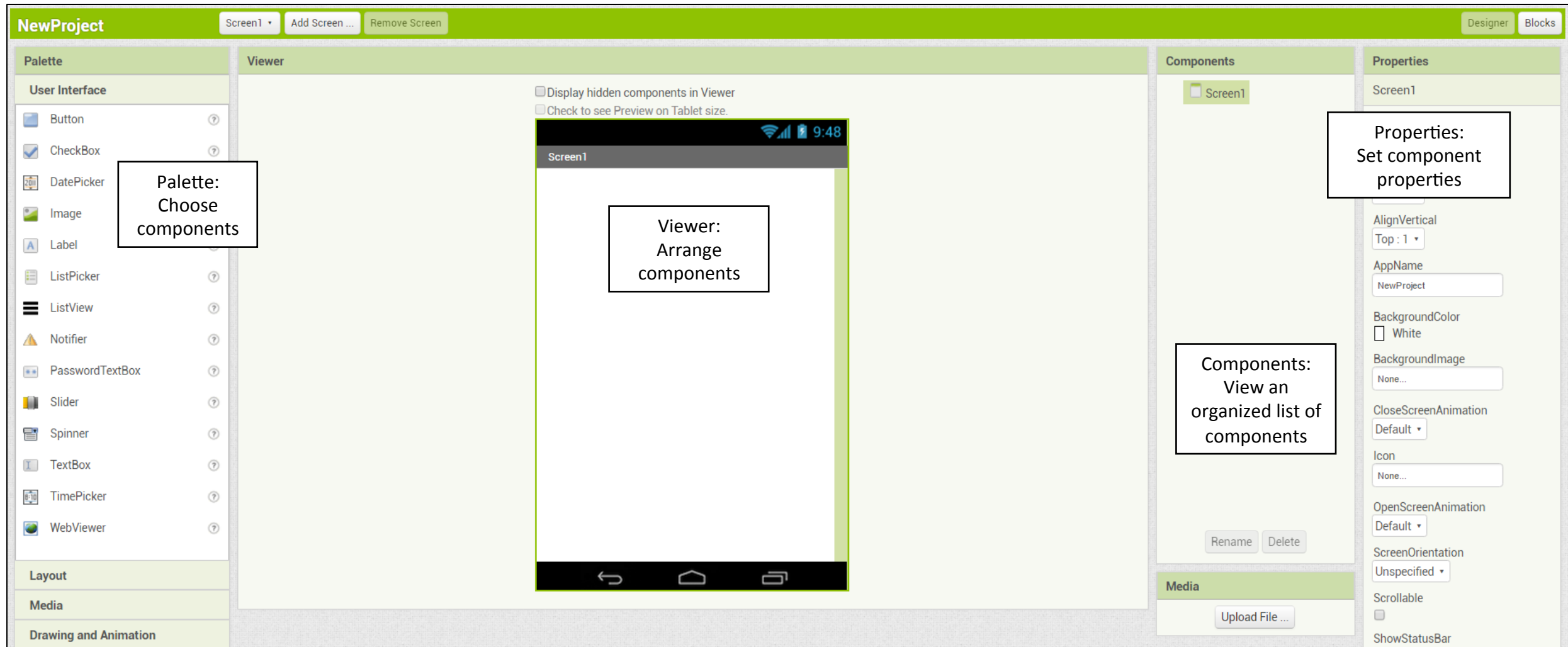
For this tutorial
you can call
your app
“MeetTim”

...give it a name and click “OK” to get started!

The screenshot shows a dialog box titled 'Create new App Inventor project'. It has a text input field labeled 'Project name:' which is currently empty. At the bottom of the dialog, there are two buttons: 'Cancel' and 'OK'. The 'OK' button is highlighted with a red rectangular border.



Step 3: Familiarize yourself with the designer window

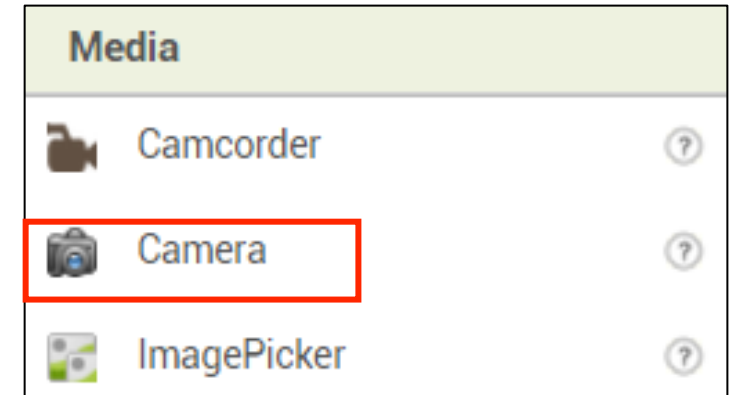
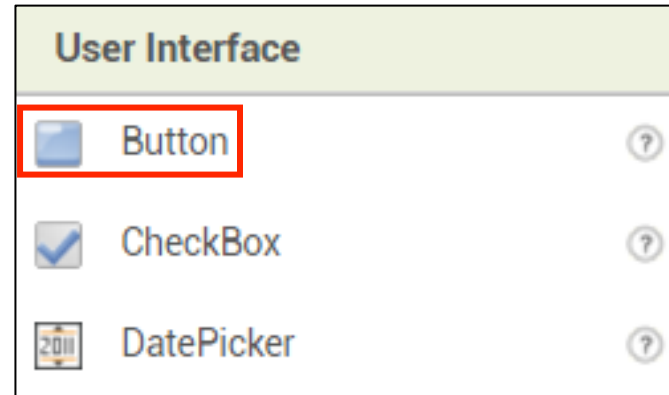
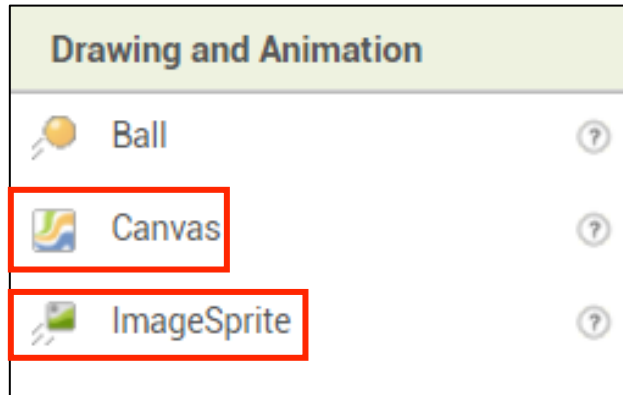


The screenshot shows the MIT App Inventor Designer interface. The top bar includes 'NewProject', 'Screen1', 'Add Screen ...', 'Remove Screen', and tabs for 'Designer' and 'Blocks'. The interface is divided into four main sections:

- Palette:** Located on the left, it contains categories like 'User Interface', 'Layout', 'Media', and 'Drawing and Animation'. The 'User Interface' category is expanded, showing a list of components: Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, TextBox, TimePicker, and WebView. A callout box labeled 'Palette: Choose components' points to this list.
- Viewer:** The central area shows a preview of the app screen. It includes a status bar at the top with signal, battery, and time (9:48) indicators. Below the status bar is a header labeled 'Screen1'. The main content area is empty. A callout box labeled 'Viewer: Arrange components' points to this area. Above the viewer, there are checkboxes for 'Display hidden components in Viewer' and 'Check to see Preview on Tablet size'.
- Components:** Located on the right, it shows a list of components currently on the screen. In this case, only 'Screen1' is listed. A callout box labeled 'Components: View an organized list of components' points to this list. Below the list are 'Rename' and 'Delete' buttons.
- Properties:** Also on the right, it shows the properties for the selected component, 'Screen1'. Properties include 'AlignVertical' (set to 'Top : 1'), 'AppName' (set to 'NewProject'), 'BackgroundColor' (set to 'White'), 'BackgroundImage' (set to 'None...'), 'CloseScreenAnimation' (set to 'Default'), 'Icon' (set to 'None...'), 'OpenScreenAnimation' (set to 'Default'), 'ScreenOrientation' (set to 'Unspecified'), 'Scrollable' (checked), and 'ShowStatusBar' (checked). A callout box labeled 'Properties: Set component properties' points to this section.

Step 4: Add components!

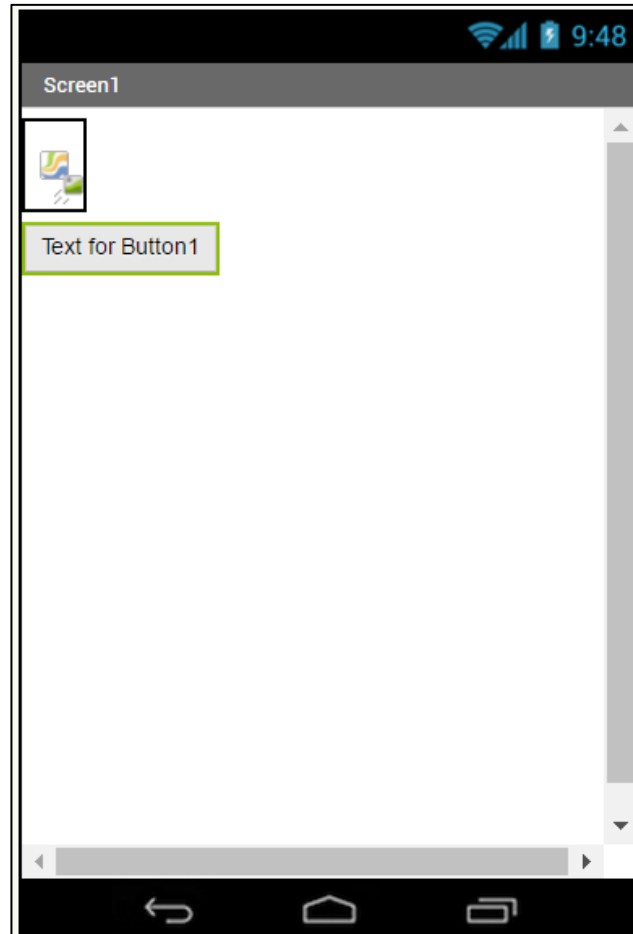
To build this app you will need four components—a canvas, image sprite, button, and camera. Find these components in the Palette and drag and drop one of each onto the Viewer.



Components are the building blocks of App Inventor apps!



Your screen should now look like this:

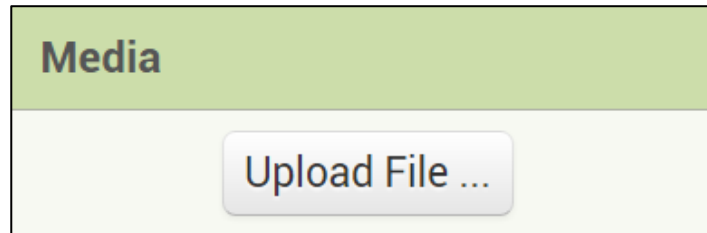


Note that no matter where you drop the Image Sprite, it will end up inside the Canvas

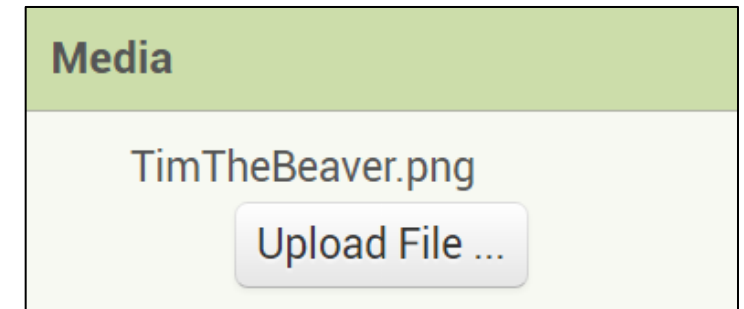


Step 5: Upload media files

To complete this app you will need to download a picture of Tim from [here](#). Then you will need to upload it to the App Inventor server by clicking the upload file button under “Media”



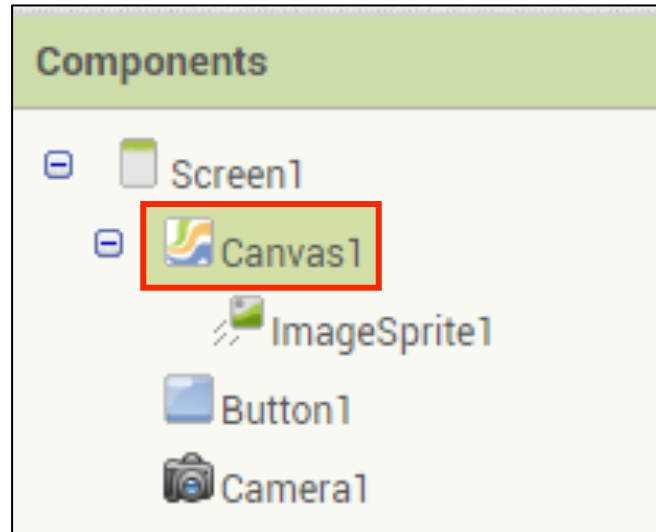
Before upload



After upload

Step 6: Set properties

Now we will change some component properties to start truly building our app! To view and change a component's properties, find it in the "Components" list and click on it.



Let's start with the canvas! Find "Canvas1" in the "Components" list and click on it. Then change both its Height and Width properties to "Fill parent". This will ensure that our canvas is as big as possible.





Step 6 continued

Select "ImageSprite1" and set the following properties:

Height to 300 pixels

Width to 225 pixels

Picture to TimTheBeaver.png

X to 91

Y to 60

Z to 1.0

Whew! There's a lot to do on this page. Double-check to make sure you don't miss anything!



Properties

ImageSprite1

Enabled ☒

Heading 0

Height 300 pixels...

Width 225 pixels...

Interval 50

Picture TimTheBeaver1.png...

Rotates ☒

Speed 0.0

Visible ☒

X 91

Y 60

Z 1.0

Properties

Button1

BackgroundColor Default

Enabled ☒

FontBold ☐

FontItalic ☐

FontSize 14.0

FontTypeface default ▾

Height Automatic...

Width Automatic...

Image None...

Shape default ▾

ShowFeedback ☒

Text Camera

TextAlignment center : 1 ▾

TextColor Default

Visible ☒

Next, select "Button1" and set the Text property to "Camera"

Properties

Screen1

AboutScreen

AlignHorizontal Center : 3 ▾

Finally, select "Screen1" and change the "AlignHorizontal" property to "Center"



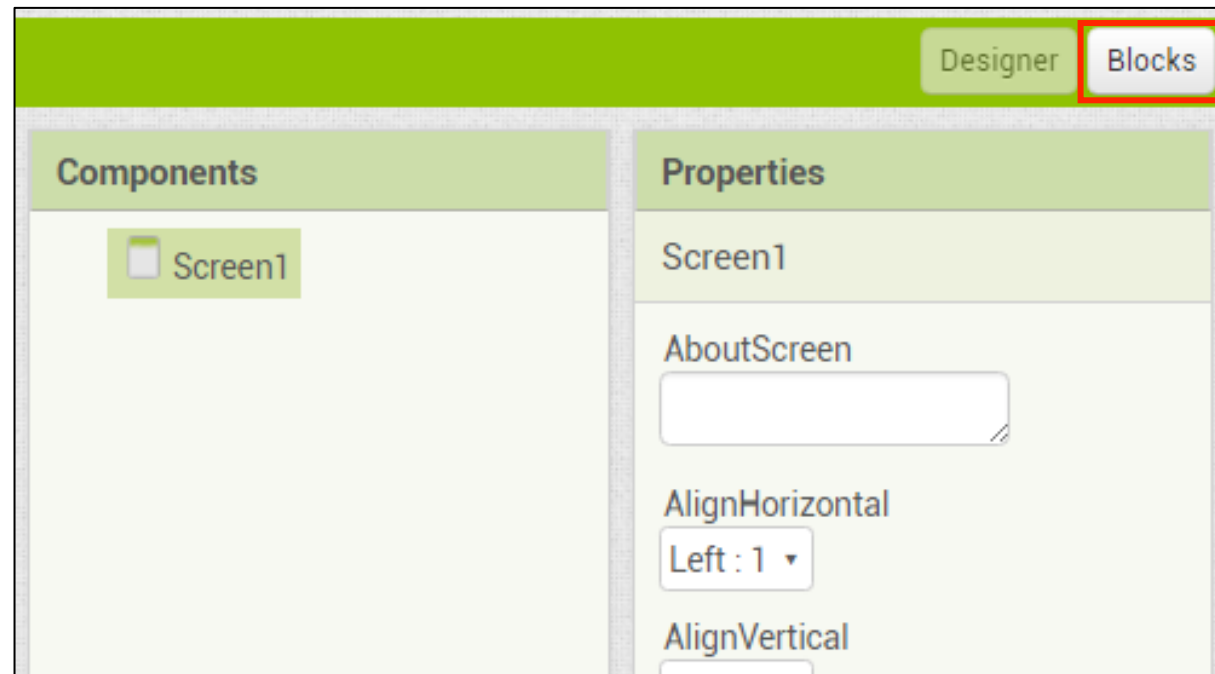
Your screen should now look like this:





Step 7: Switch to the blocks window to write code!

Now that all components have been added to the app, we will write code to tell the app what to do with them! To do so, switch to the blocks window by clicking the “Blocks” button in the upper right corner.





Step 7 continued: Get to know the blocks window

TakePictureWithTim Screen1 Add Screen ... Remove Screen Designer Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - Canvas1
 - Tim
 - CameraButton
 - Camera1
- Any component

Rename Delete

Media

- TimTheBeaver.png
- TimTheBeaver1.png
- Upload File ...

Viewer

Built-in blocks:
These are always available and handle things like math, text logic, and control

Component blocks:
These correspond to the components you've added to your app

An example of two assembled blocks

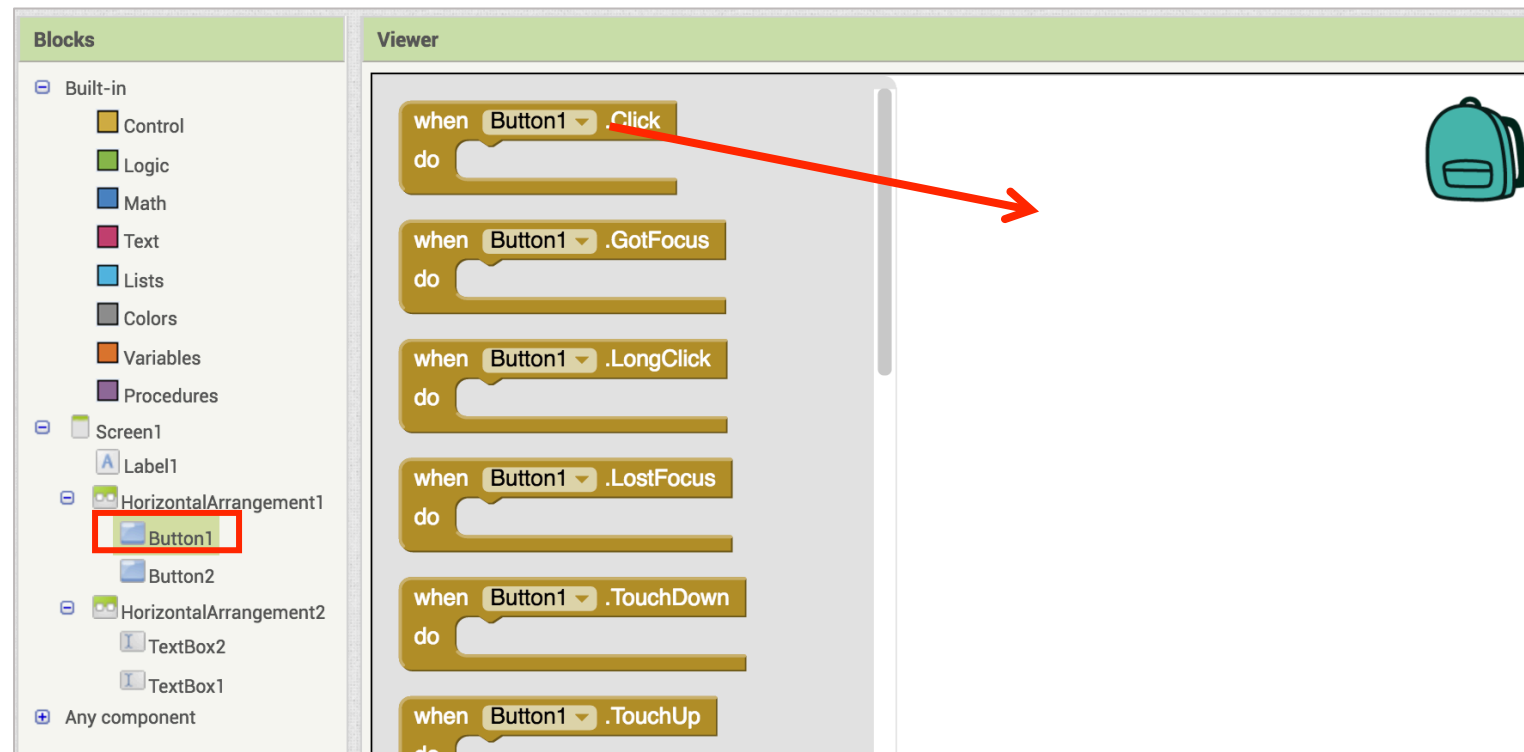
```
when CameraButton Click
do call Camera1 TakePicture
```

Viewer:
Where you assemble the blocks into a program

0 0
Show Warnings

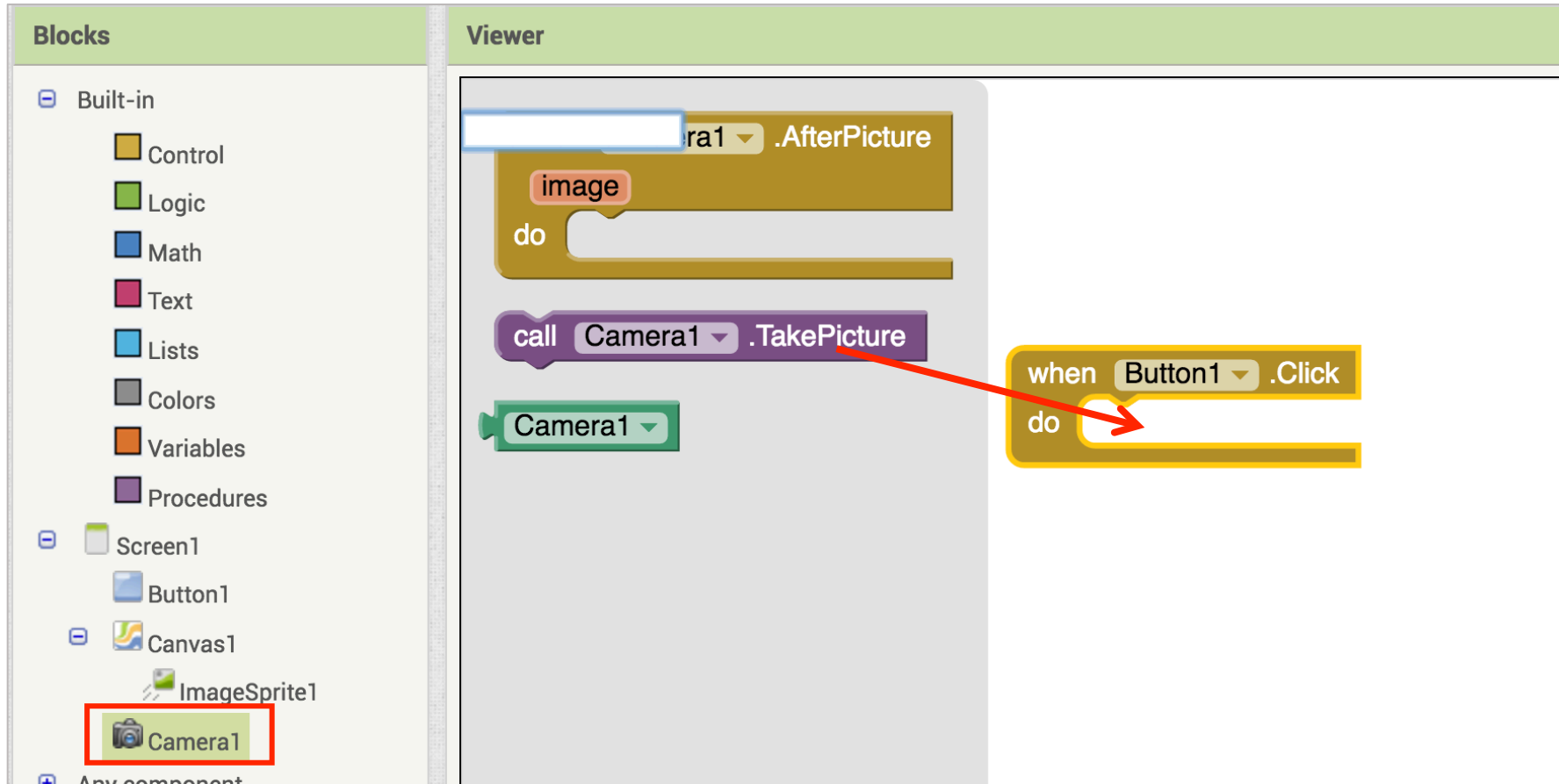
Step 8: Start coding!

When the Camera button is clicked, your phone's camera should open.
We will write the code to make this happen now!



Find Button1 under Screen1, click on it, and drag out a “when Button1.Click” block

Step 8 cont'd



Click on Camera1 this time and drag out a “call Camera1.TakePicture” block.
Lock it into in the Button1.Click block!



Step 9: Testing!



Great job! You just wrote code in App Inventor! But does your code do what we want it to? To find out, we're going to have to learn how to test our app...

Step 9 continued: Connect to your phone

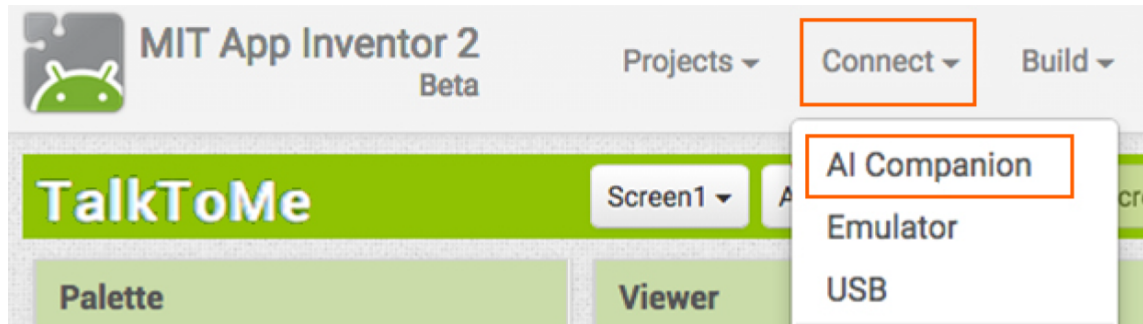
In order to test your app, you will need an Android phone with the MIT AI2 Companion app installed. To download the Companion from the app store, scan the QR code below or search directly for “MIT AI2 Companion” on the Google Play Store, <https://play.google.com/store>.



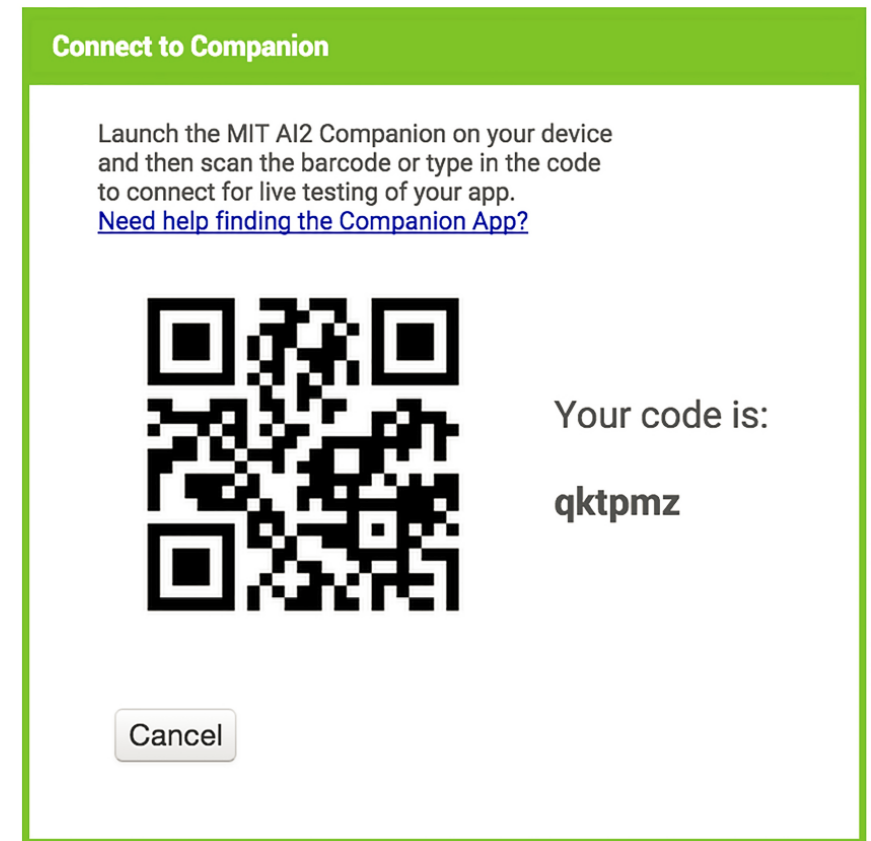
NOTE: If you do not have an android phone, or if you are unable to download the Companion app, you can still use App Inventor using an emulator. Visit: <http://appinventor.mit.edu/explore/ai2/setup.html> and follow the instructions under Option 2.

Step 9 continued

To connect to the AI2 Companion app, first choose “AI Companion” from the “Connect” drop down menu in the App Inventor site.

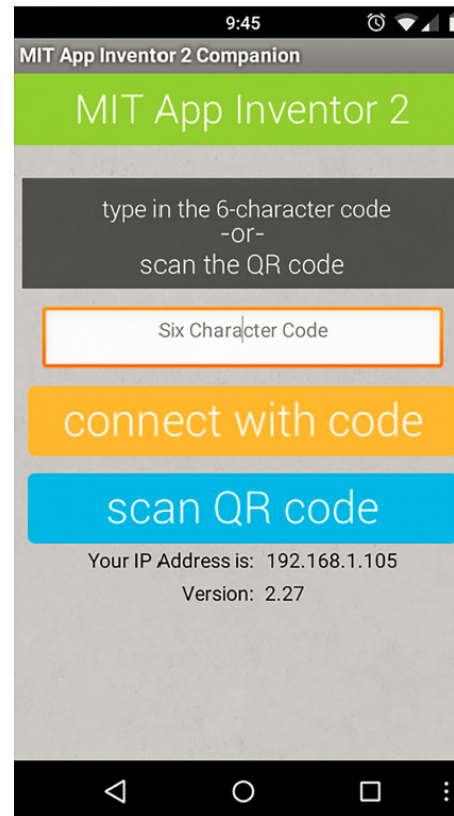


A QR code and 6-letter code will pop up.



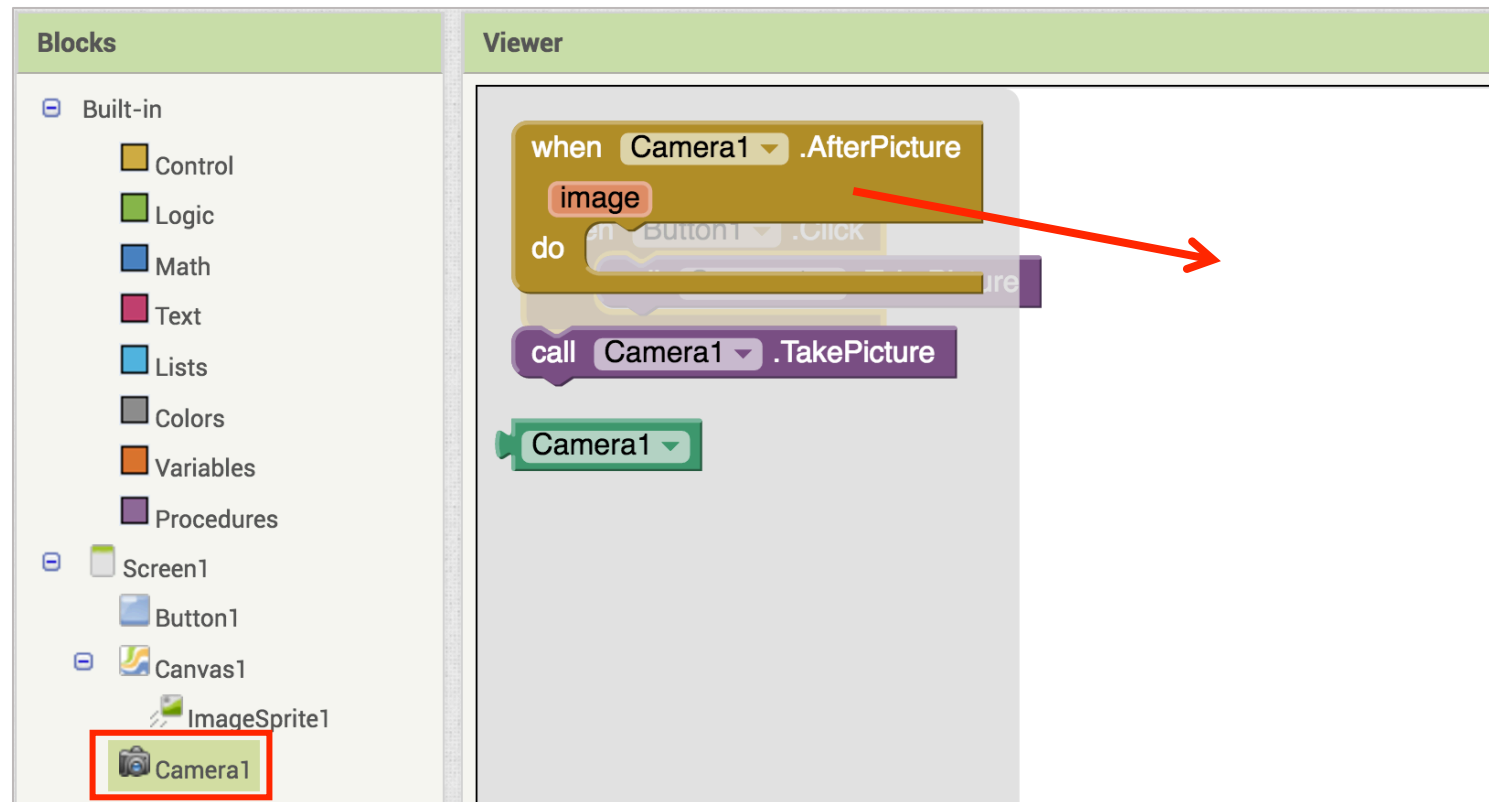
Step 9 cont'd: Open the companion app

Open the companion app. You can then either input the 6-letter code or scan the QR code to connect.



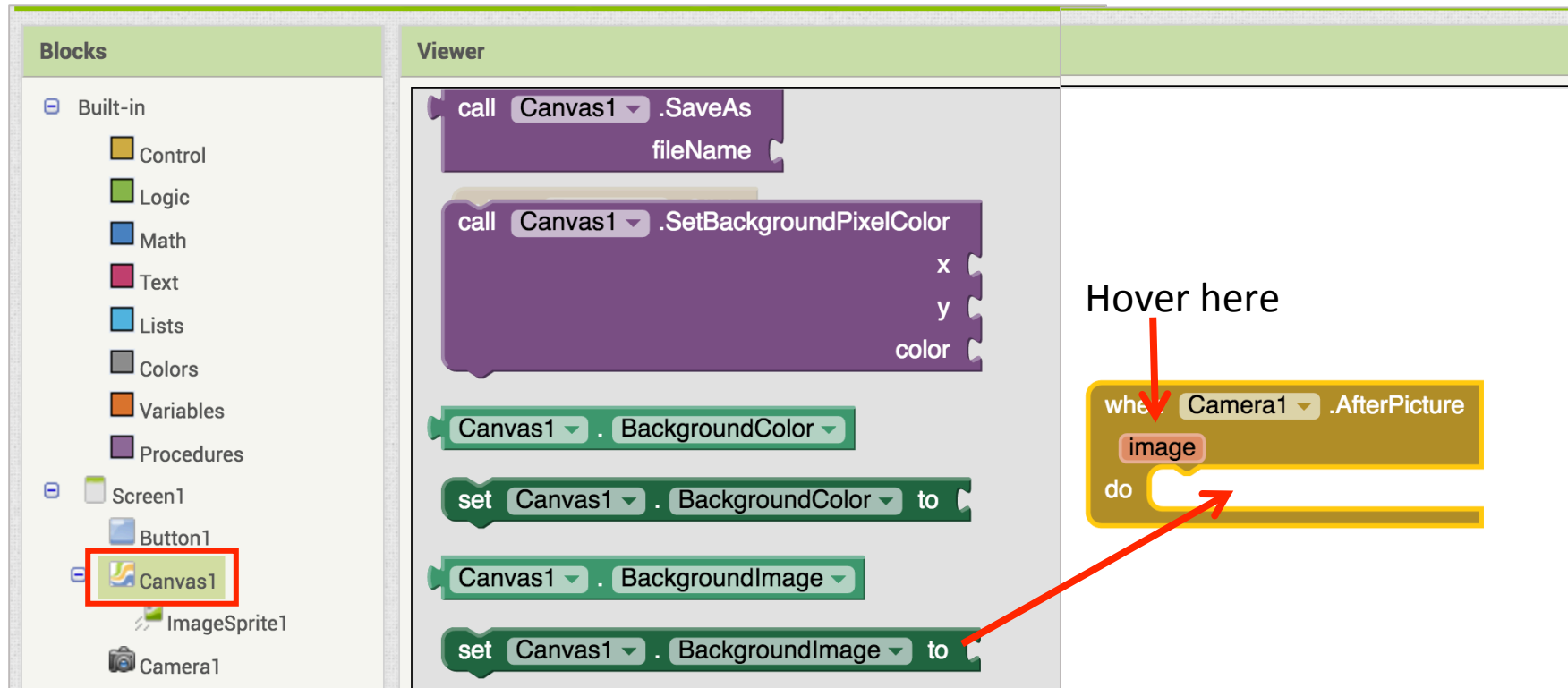
Step 10: More programming!

Awesome! We've created an app that opens the phone camera and lets us take a picture. But we're not done yet! Once a picture is taken, the app should set it to be the Canvas background.



Find Camera1 in the blocks menu under Screen1. Click on it and drag a “when Camera1.AfterPicture” block onto the workspace

Step 10 continued



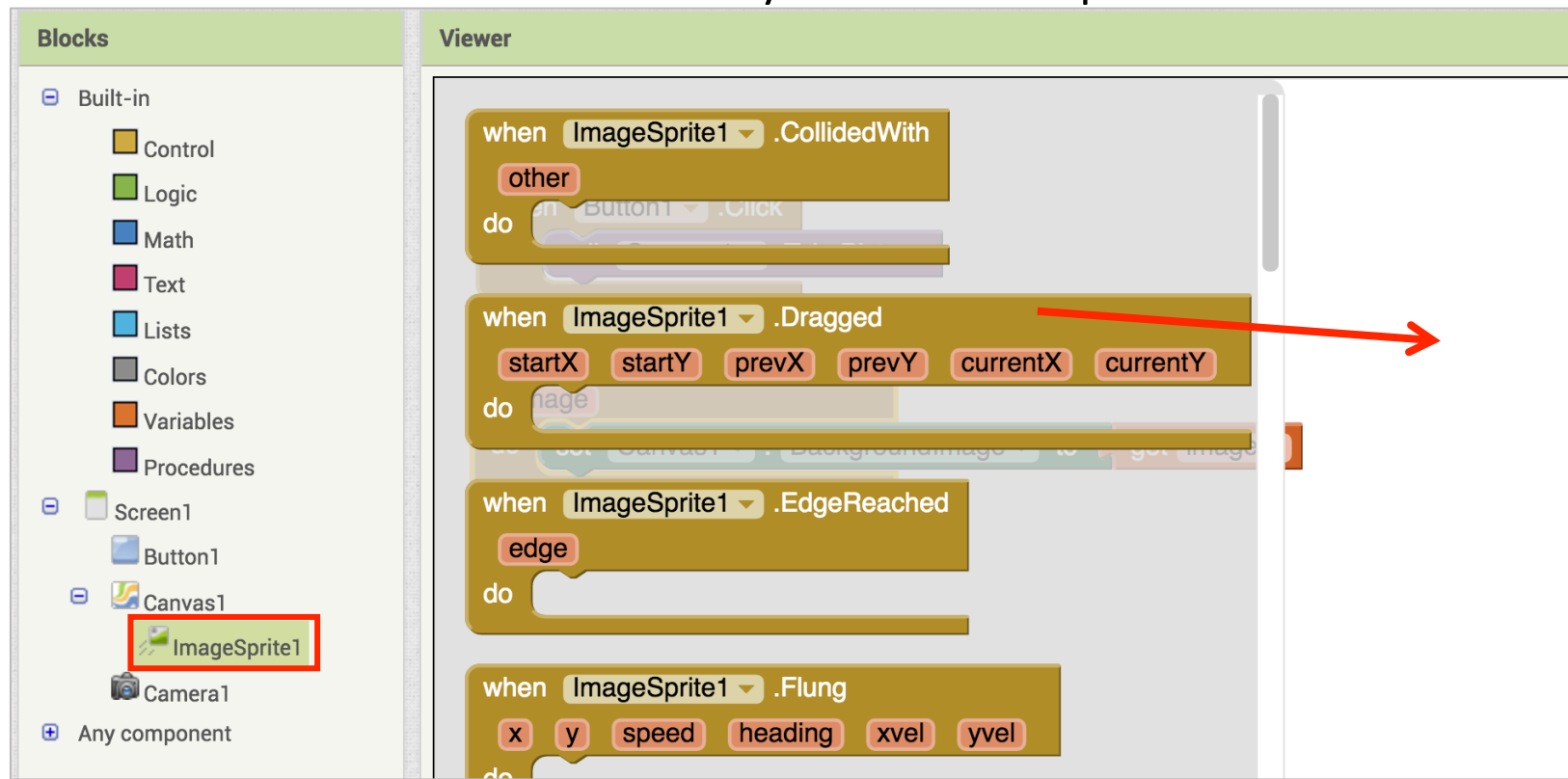
Now find Canvas1, also under Screen1. Click on it and drag out a “set Canvas1.BackgroundImage” block. Click it into place under the “when” block. Then hover your mouse over the orange “image” box on the “when” block and drag out a “get image” block. Snap it into place!



Wow! You're doing great so far. But what about me? I'd like to be able to move around on the screen

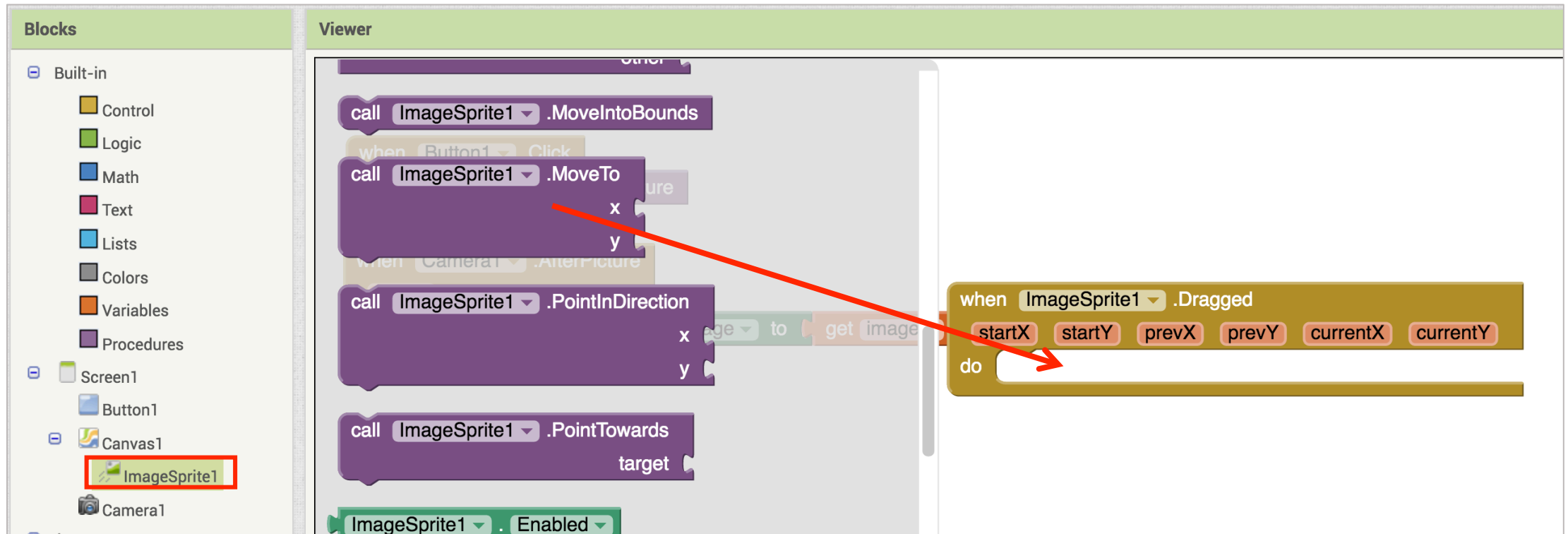
Step 10 continued: Moving Tim

We would like Tim to move when he is dragged, so that the app user can place him where they'd like in their picture!



Find ImageSprite1 under Canvas1, click on it and drag out a “when ImageSprite1.Dragged” block

Step 10 continued



The screenshot displays the MIT App Inventor interface. On the left, the 'Blocks' pane shows a list of built-in blocks categorized by Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Under the 'ImageSprite1' category, the 'ImageSprite1' block is highlighted with a red rectangle. On the right, the 'Viewer' pane shows a sequence of blocks for 'ImageSprite1'. The blocks are: 'call ImageSprite1.MoveIntoBounds', 'call ImageSprite1.MoveTo' (with 'x' and 'y' inputs), 'call ImageSprite1.PointInDirection' (with 'x' and 'y' inputs), and 'call ImageSprite1.PointTowards' (with a 'target' input). A red arrow points from the 'MoveTo' block in the Viewer to the 'do' field of the 'when ImageSprite1.Dragged' block on the right.

Under ImageSprite1 again, find a “call ImageSprite1.MoveTo” block and lock it into place in the ImageSprite1.Dragged block.



Step 10 continued

The screenshot shows the MIT App Inventor interface. On the left is the 'Blocks' pane with a 'Built-in' category expanded, showing various block categories like Control, Logic, Math, Text, Lists, Colors, Variables, and Procedures. Below these are specific components: Screen1, Canvas1, ImageSprite1 (highlighted), Button1, and Camera1. On the right is the 'Viewer' pane, which displays a code block. The code block is a yellow 'when' block triggered by 'ImageSprite1 .Dragged'. It contains a 'do' block with a 'call ImageSprite1 .MoveTo' block. The 'MoveTo' block has two input fields, 'x' and 'y'. Red arrows point from the 'currentX' and 'currentY' variable boxes in the 'when' block to the 'x' and 'y' input fields of the 'MoveTo' block, respectively.

Hover your mouse over the “currentX” box in the “when” block.
Grab a “get CurrentX” block and snap it into place next to the “x” in the “call” block;
then grab a “get CurrentY” and snap it into place next to the “y” in the “call” block.

Step 11: Testing and debugging!

Awesome! You're all done programming this app. Now connect to the MIT AI2 Companion app to make sure everything is working properly.

Remember, your app should:

- Open the phone camera when you click the button labelled "Camera"
- Set the background of the canvas to the picture you take
- Include an image of Tim that you can drag around to place in your picture!

If you want to keep building, check out ways to extend this app!





MIT App Inventor



Thanks for coding with us!
You've done a great job. Check
out more tutorials at
[http://appinventor.mit.edu/
explore/hour-of-code.html](http://appinventor.mit.edu/explore/hour-of-code.html)