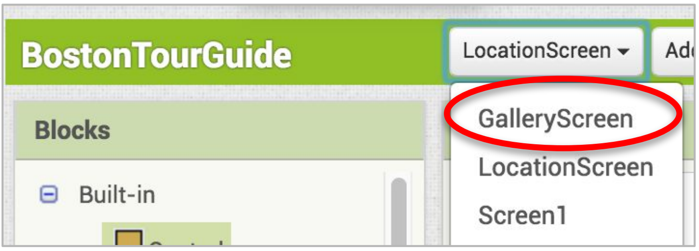
TOUR GUIDE: GALLERY SCREEN

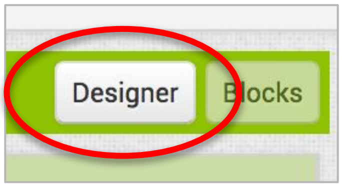
The GalleryScreen will allow users to take pictures of the sites to add to a Gallery!

GALLERY SCREEN

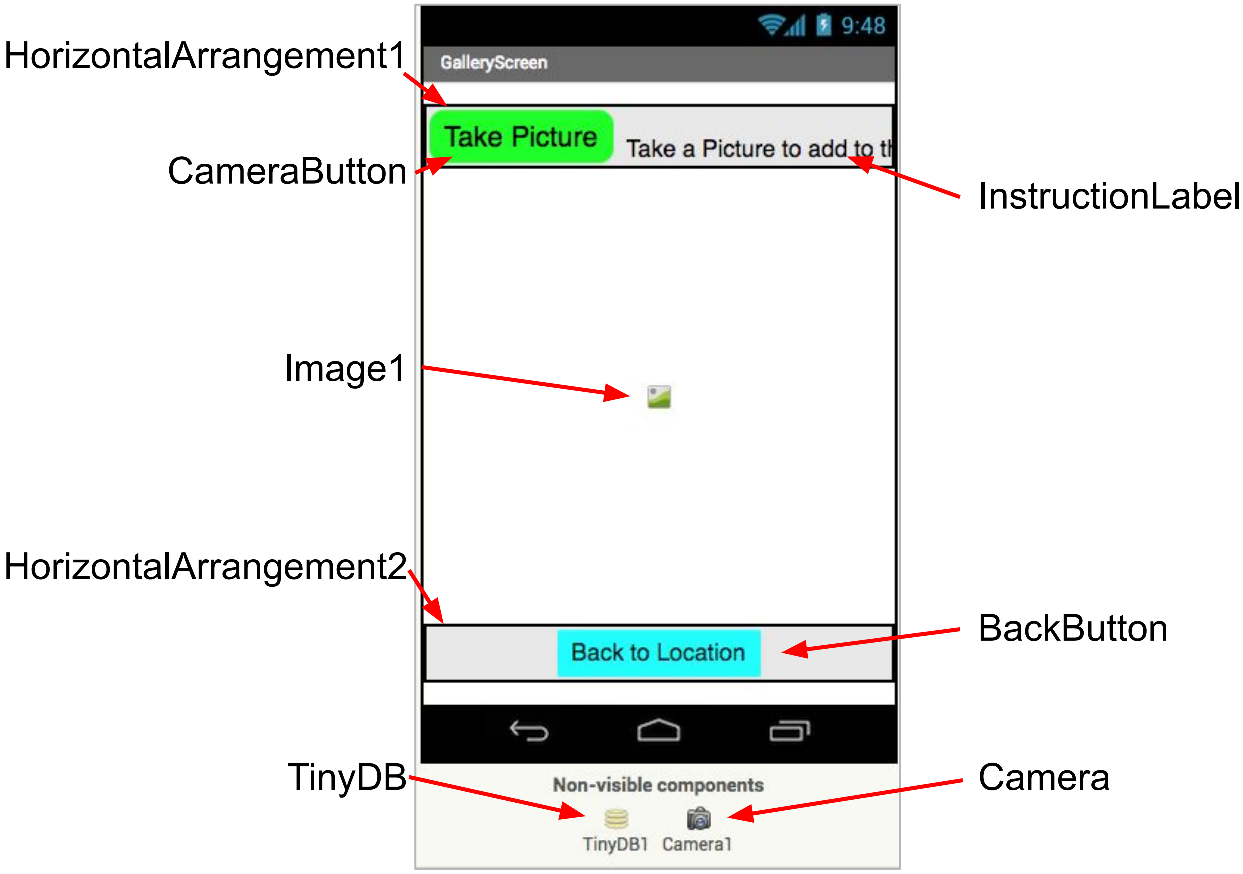
1 Go to the “**GalleryScreen**”.



2 Switch to the Designer.



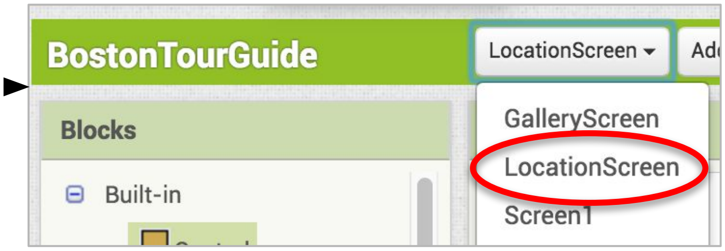
3 With your partner, look at the user interface and see if you can identify what each component does. See the screen layout below.



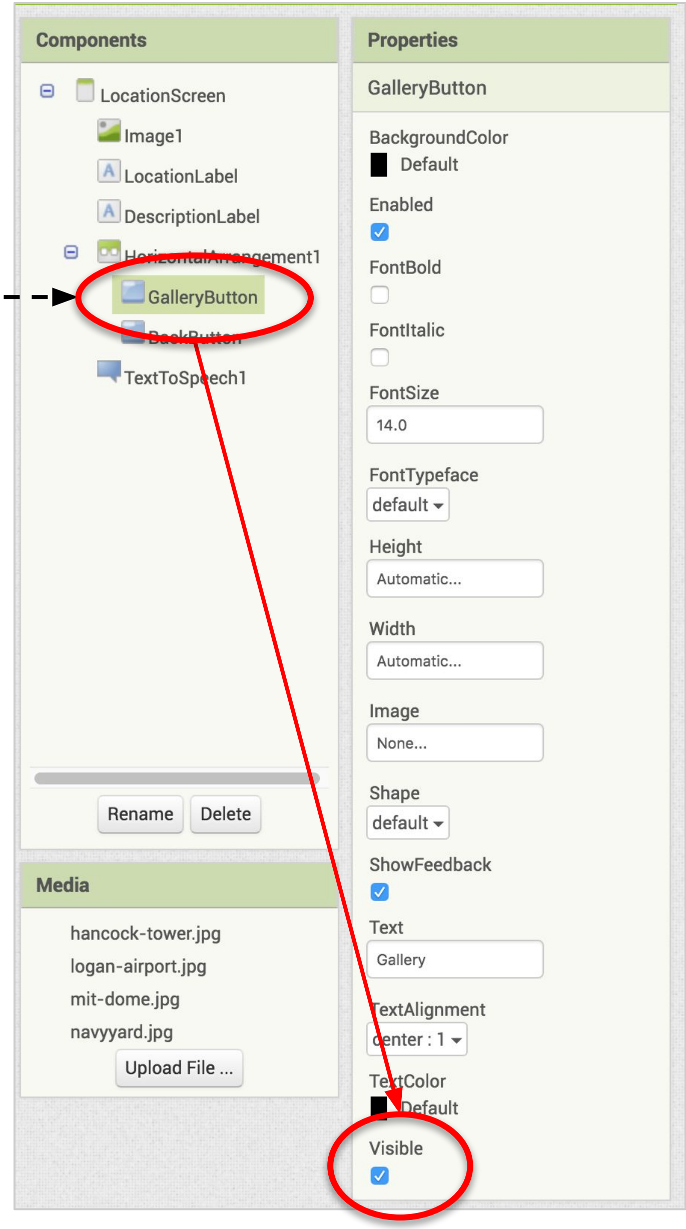
GALLERY BUTTON

Now you'll go back to **LocationScreen** and add the button and code to open the **GalleryScreen.**

4 Go back to the "**LocationScreen**".



5 Select **GalleryButton** in the Components panel, and click on the *Visible* checkbox to make it visible.



GalleryButton

BackgroundColor
■ Default

Enabled
☑

FontBold
☐

FontItalic
☐

FontSize
14.0

FontTypeface
default ▾

Height
Automatic...

Width
Automatic...

Image
None...

Shape
default ▾

ShowFeedback
☑

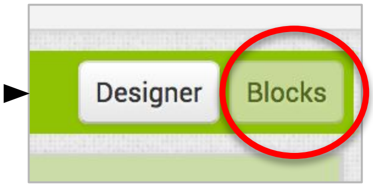
Text
Gallery

TextAlignment
center : 1 ▾

TextColor
■ Default

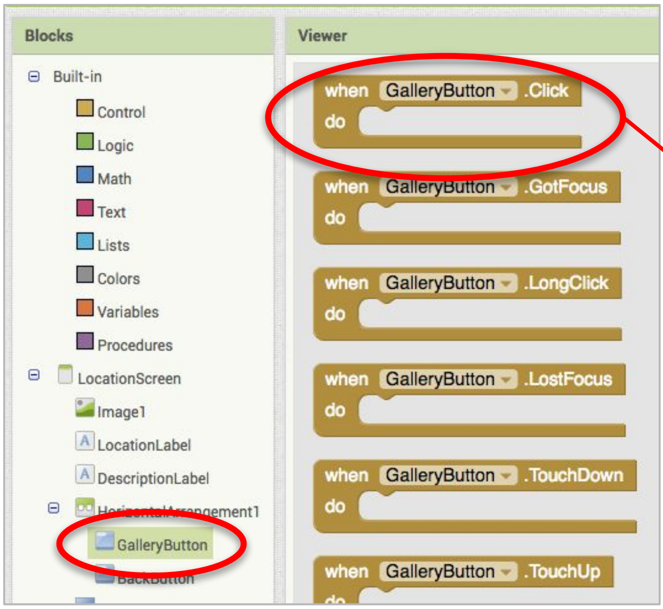
Visible
☑

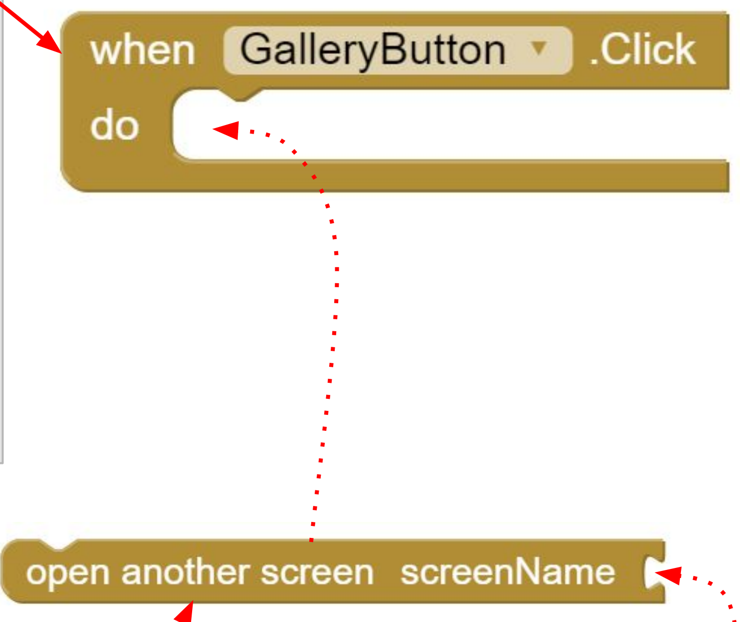
6 Switch to the Blocks Editor.



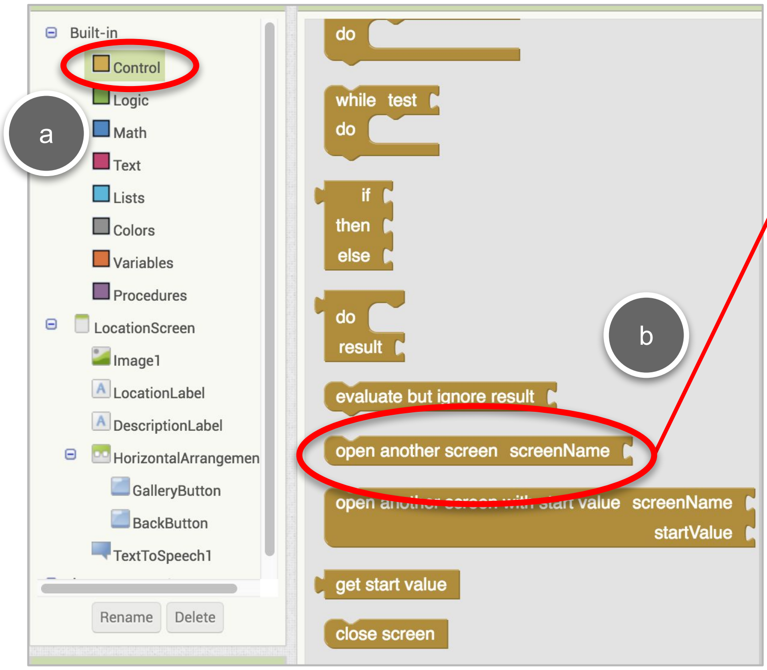
GALLERY BUTTON

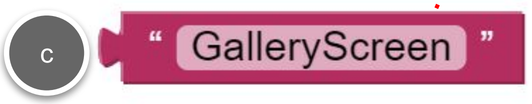
7 Drag out a **GalleryButton.Click event block.**





8 In **GalleryButton.Click** event, open the **GalleryScreen**.

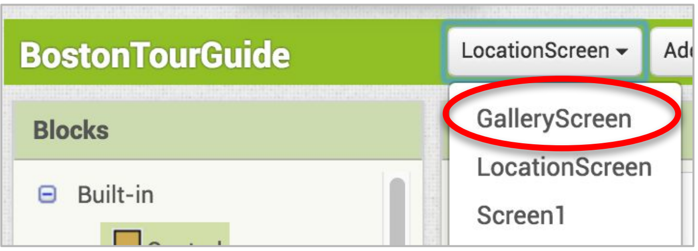




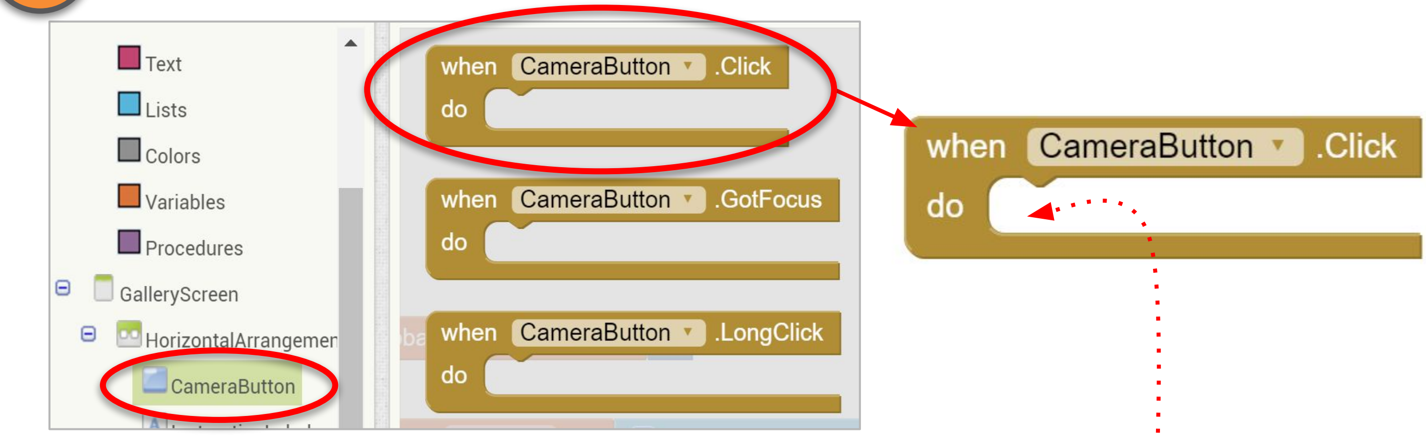
This time, you'll open **GalleryScreen**, without any startValue

TAKE PICTURES WITH THE CAMERA

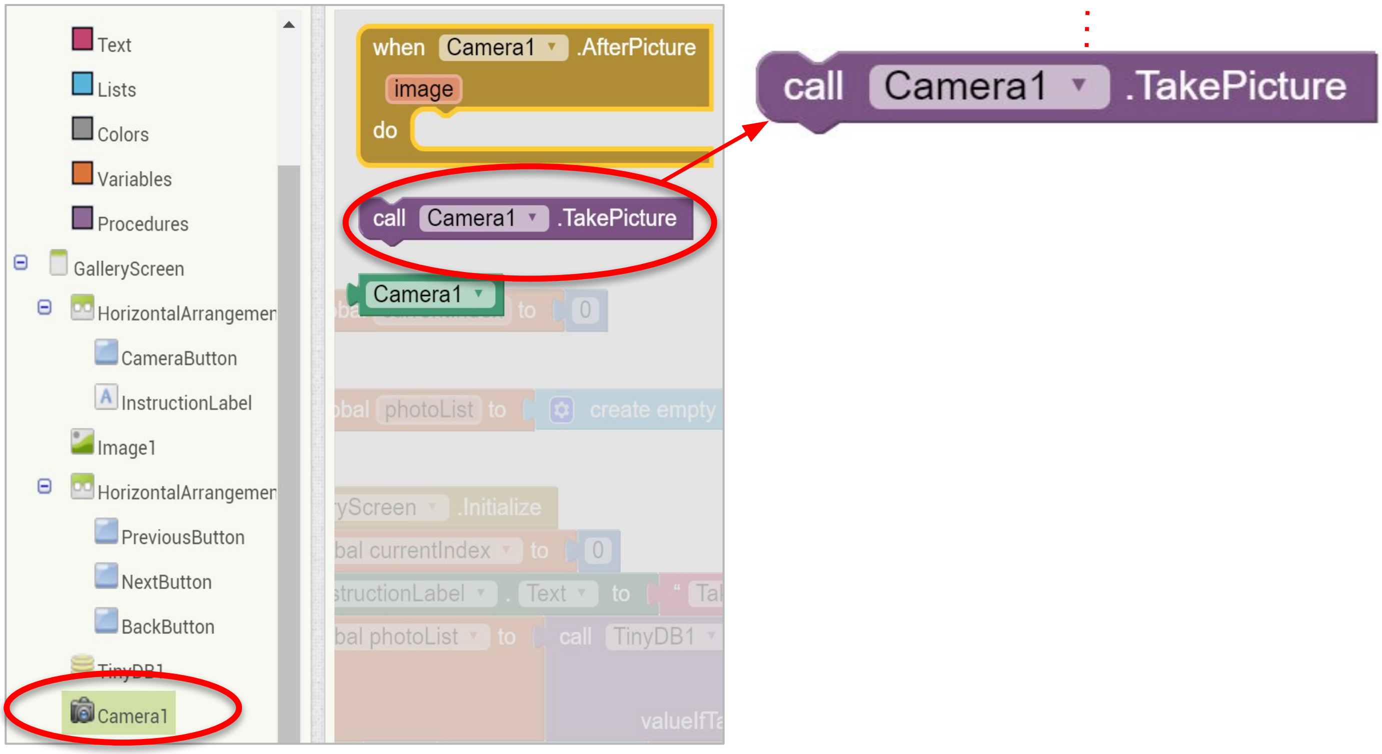
9 Now, go back to the **GalleryScreen**.



10 Drag out a **CameraButton.Click** event.

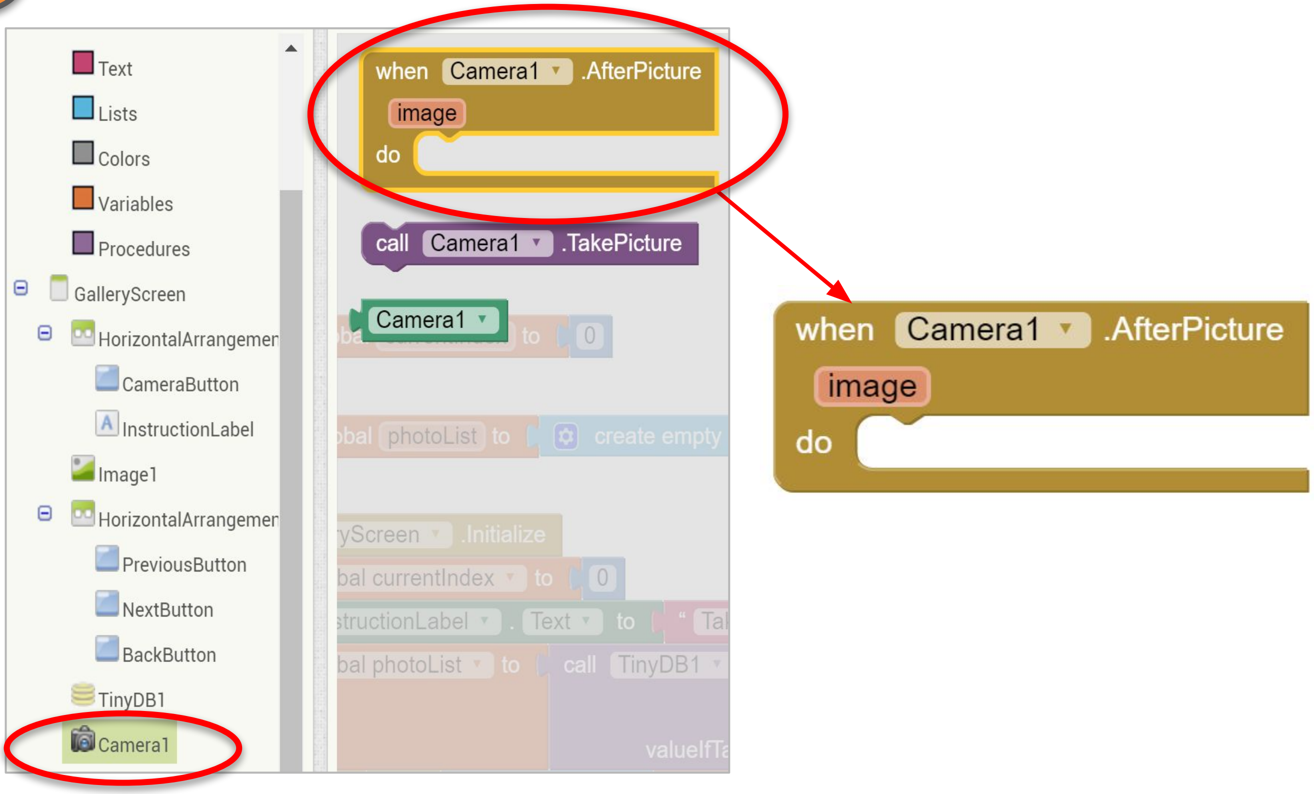


11 When the button is clicked, take the picture!

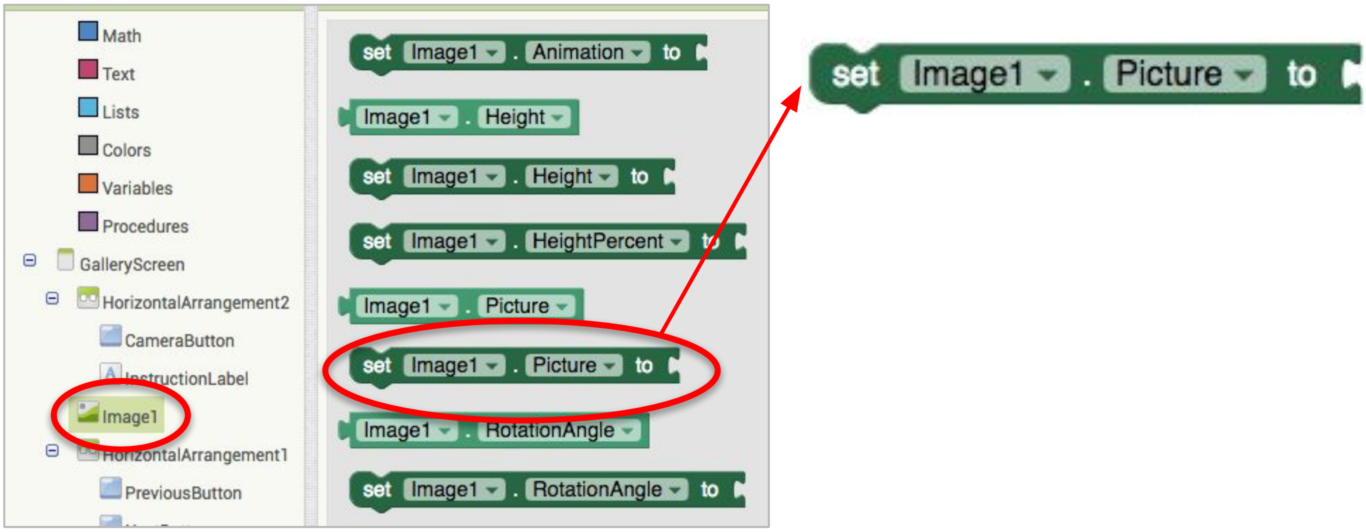


DISPLAY PICTURE

12 After a picture is taken, the **Camera.AfterPicture** event is triggered.

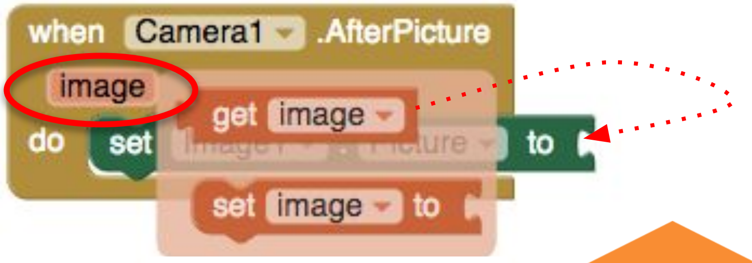


13 Set the **Image.Picture** to the image the camera just took.



AFTER PICTURE

14 Hover over image and snap it into **set Image1.Picture.**



15 Test the Camera feature with the MIT AI2 Companion.

Take a picture, and see that the image changes.

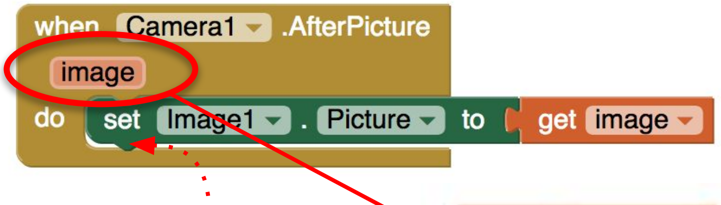
Note: If you take a second picture, it will overwrite the first.

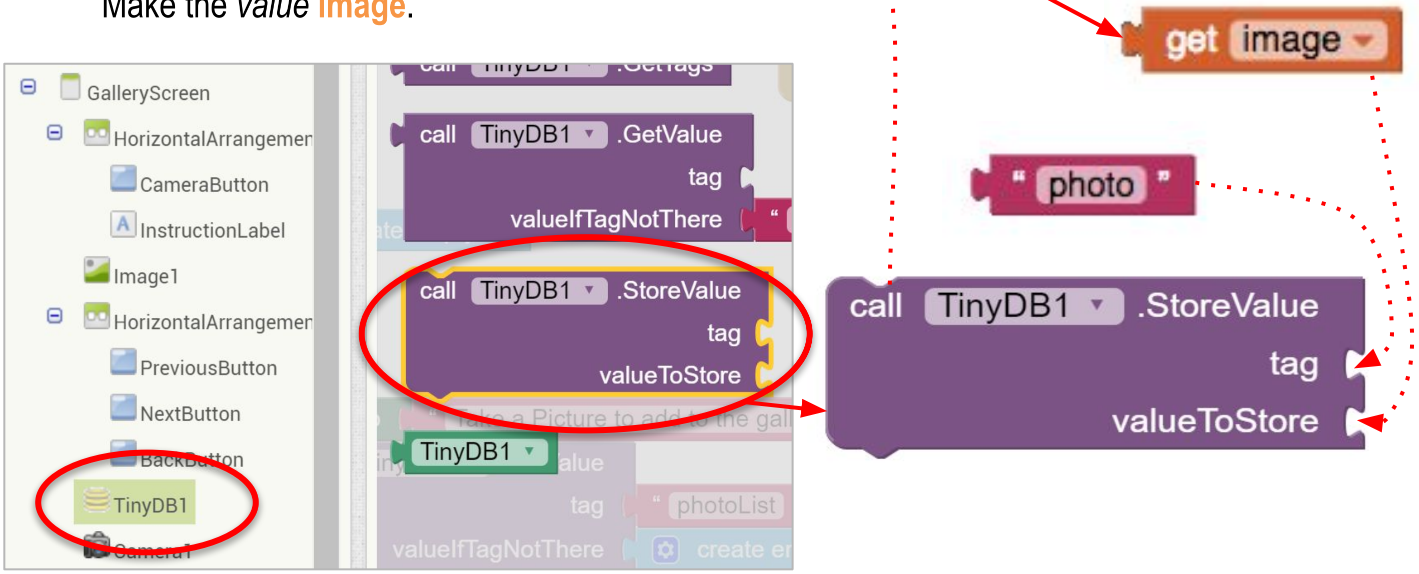
STORE PICTURE IN TINYDB

To make sure the picture is saved for the next time someone uses the app, you need to store it in **TinyDB**. Remember, **TinyDB** stores information persistently, so it will always be saved.

When a picture is taken, store it in TinyDB.

16 Drag out a **TInyDB1.StoreValue** block. Make the *tag* "**photo**". Make the *value* image.


when Camera1 .AfterPicture
image
do set Image1 . Picture to get image

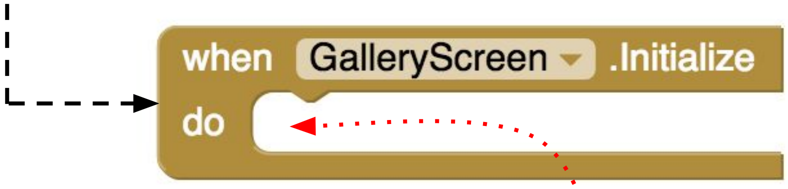


GalleryScreen
HorizontalArrangemen
CameraButton
InstructionLabel
Image1
HorizontalArrangemen
PreviousButton
NextButton
BackButton
TinyDB1
Camera1

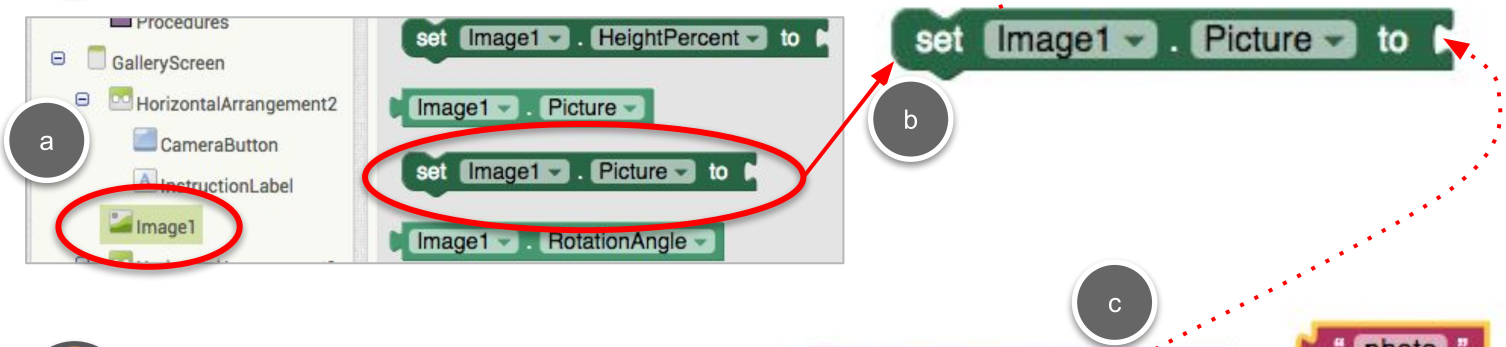
INITIALIZE SCREEN

Because you are storing the photo to **TInyDB**, you need to get it from **TinyDB** each time the screen opens. You will code that in the **GalleryScreen.Initialize** event.

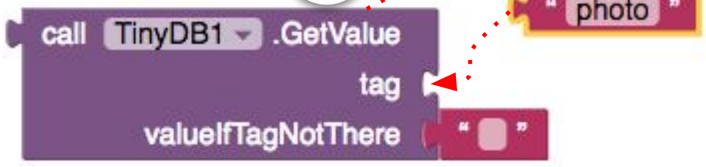
17 Drag out a **GalleryScreen.Initialize** event**.**



18 Set the **Image.Picture** to the photo stored in **TInyDB**.



19 Use “**photo**” as the *tag* again. And leave the *valueIfTagNotThere* as blank text, so no picture appears.



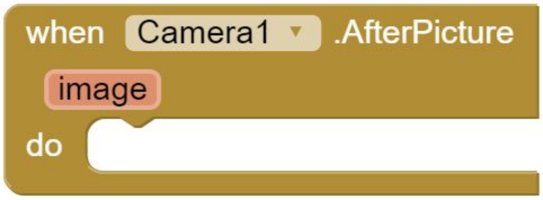
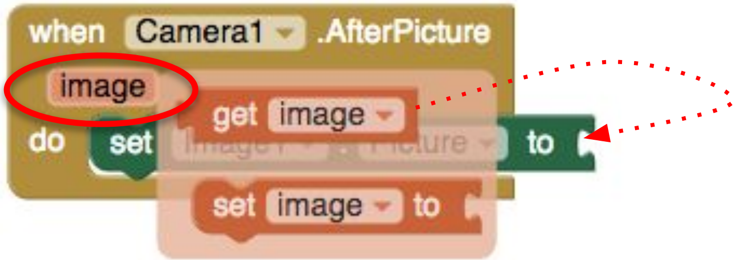
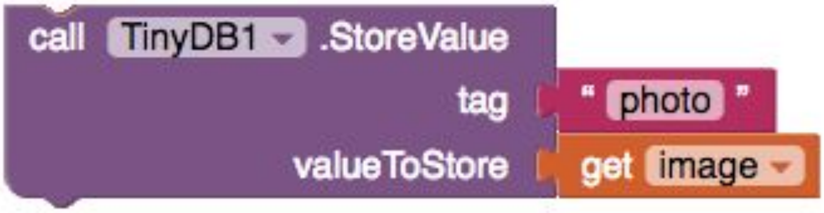
20 Here you go! Test out your app with the MIT AI2 Companion.

- Go to the **GalleryScreen**.
- Take a picture.
- Close the app and reopen it.

The picture should still be there!

COMPUTATIONAL THINKING CONCEPTS

The following are the Computational Thinking Concepts used in GalleryScreen.

Tour Guide
1. Events 
2. Variables/Naming  when Camera1 .AfterPicture image get image do set Image1 . Picture to set image to
3. Data manipulation and elementary data structures  call TinyDB1 .StoreValue tag " photo " valueToStore get image