

TOUR GUIDE: MULTIPLE PICTURES

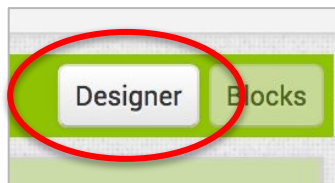
UPDATE GALLERY SCREEN



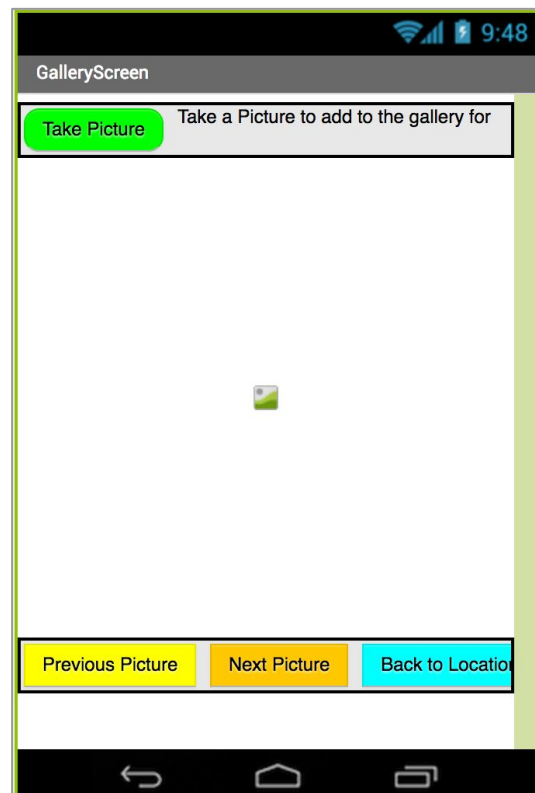
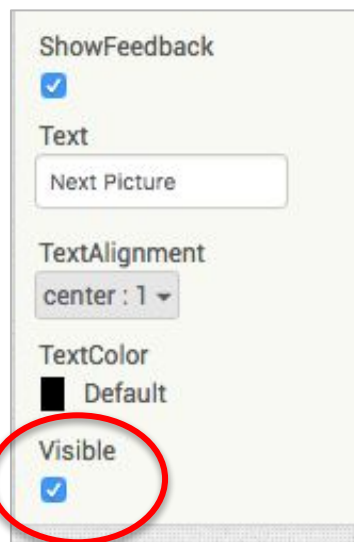
You'll add the functionality for more than one picture in the Gallery!

1 Go to the "GalleryScreen".

2 Switch to the Designer.



3 The template contains two invisible buttons, **PreviousButton** and **NextButton**. Find them in the Components panel and check their *Visible* property.



The two buttons appear next to the **BackButton**. -->

AFTER PICTURE

Make your app more versatile, by allowing users to take many pictures instead of just one! You'll use a List called **photoList** to store the images in the app, and **TinyDB** to store the list so it can be retrieved each time the user runs the app.

4 Initialize two new variables. Name them and set them as seen below.

a

b **currentIndex** points to the index of the current picture in **photoList**.

c **photoList** is the list of photos.

5 When a picture is taken, add the **image** to the **photoList**.

a

b

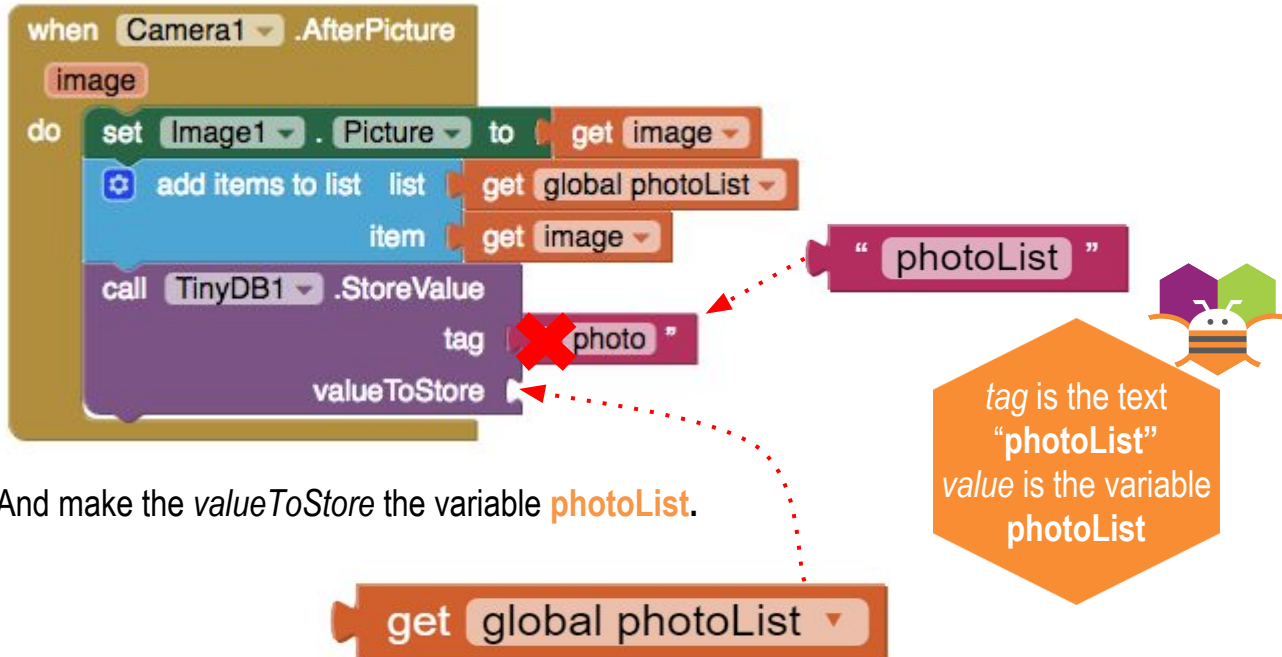
c

d

e

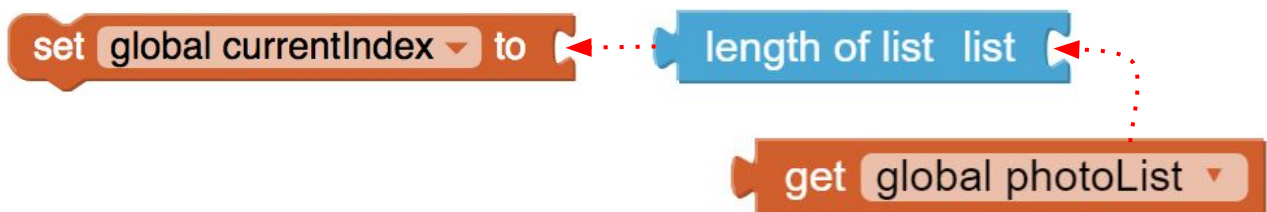
STORING THE PHOTOLIST

- 6 Change the tag for **TinyDB1.StoreValue** to “photoList”.



- 7 And make the *valueToStore* the variable **photoList**.

- 8 Update **currentIndex** to point to the end of your list, by setting it to the length of the list.

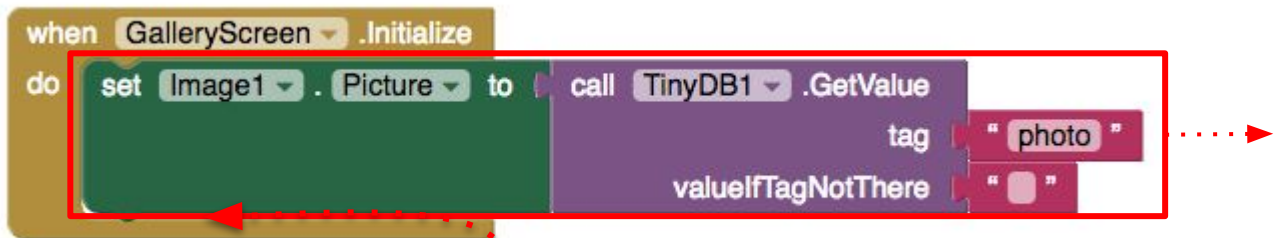


- 9 Snap that block in below the **add items to list** block.

INITIALIZE SCREEN

Because you are switching from a single photo to a list, you need to update the **GalleryScreen.Initialize** event.

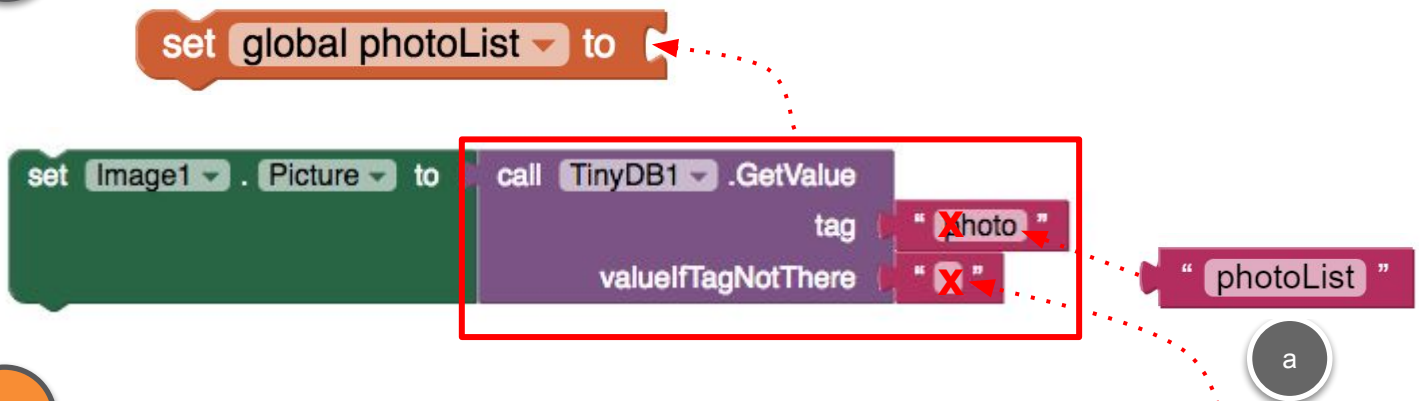
- 10 Remove the **set Image1.Picture** block, and set it aside. Don't delete it.



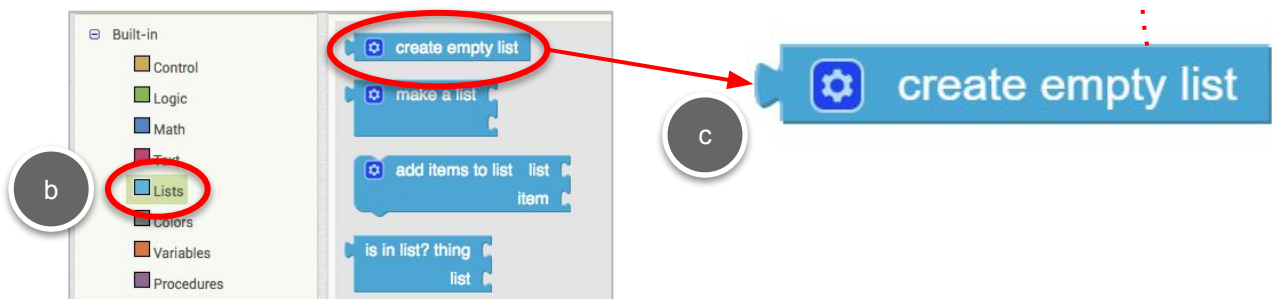
- 11 Start by setting **currentIndex** to 0.



- 12 Then add a **set photoList** block. Snap the **TinyDB.GetValue** block to that block.

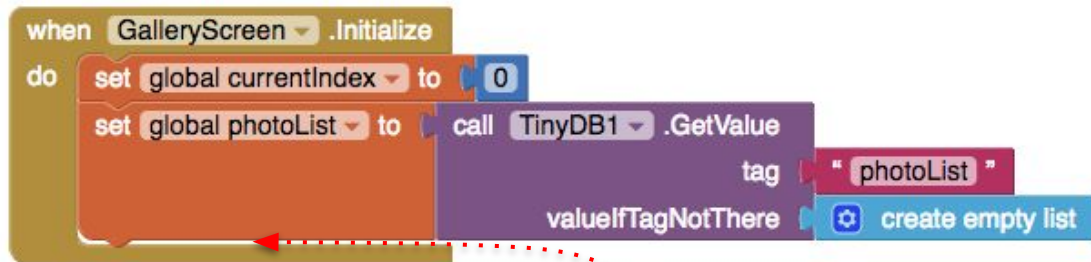


- 13 Replace "**photo**" with "**photoList**" and the blank Text string with **create empty list**.

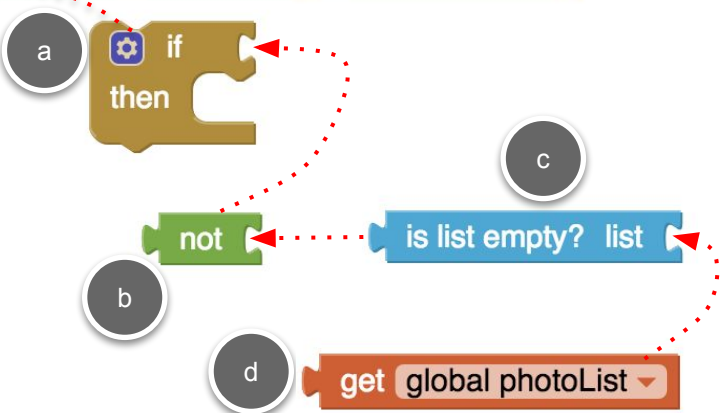


INITIALIZE SCREEN

You need to check if you have anything in your list yet.



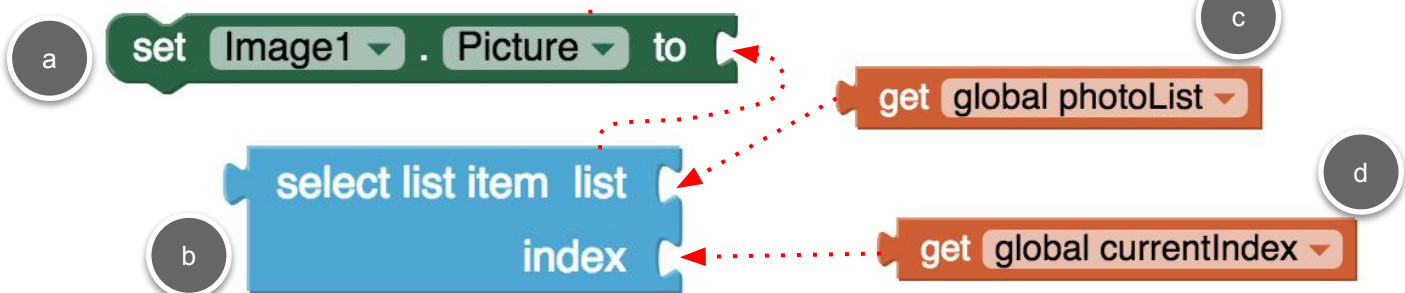
- 14 Drag out an **if-then** block.
Snap in blocks to check if the list is **not** empty, which means there are pictures in the list.



- 15 If it's not empty, point to the first photo, index 1.



- 16 And set the **Image.Picture** to the item in the list at that index.



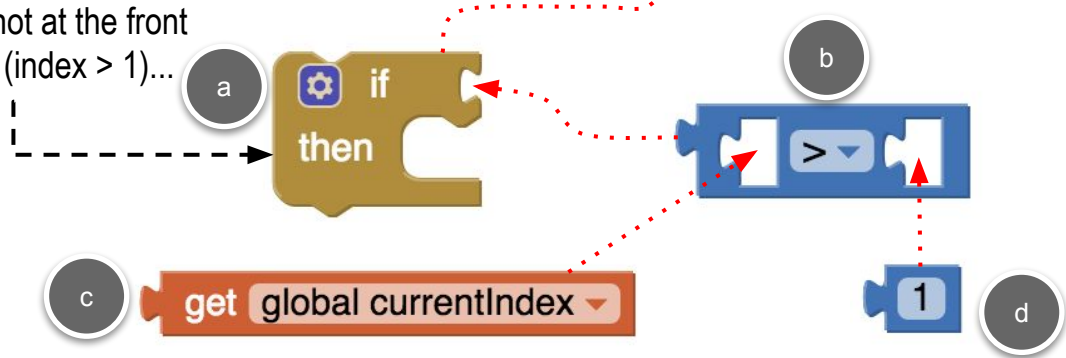
PREVIOUSBUTTON

Now code the **PreviousButton** and **NextButtons**. You need to update **currentIndex**, by either adding to go forward or subtracting to go back, in your list of photos.

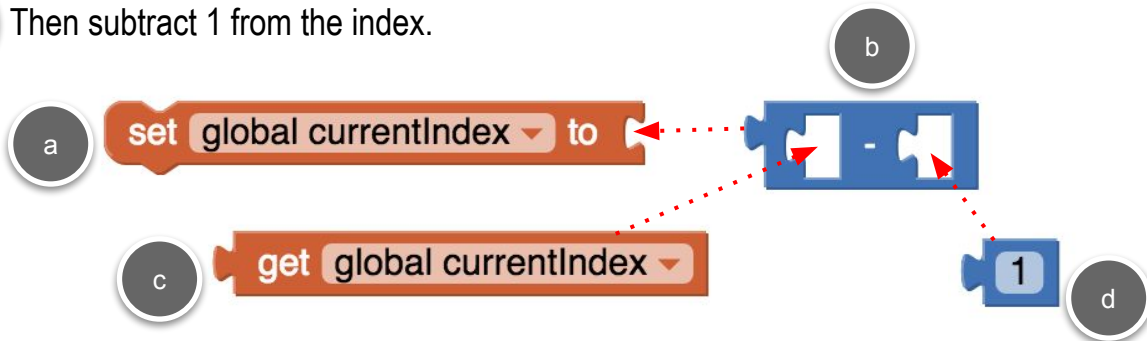
- 17 Drag out a **PreviousButton.Click** block.



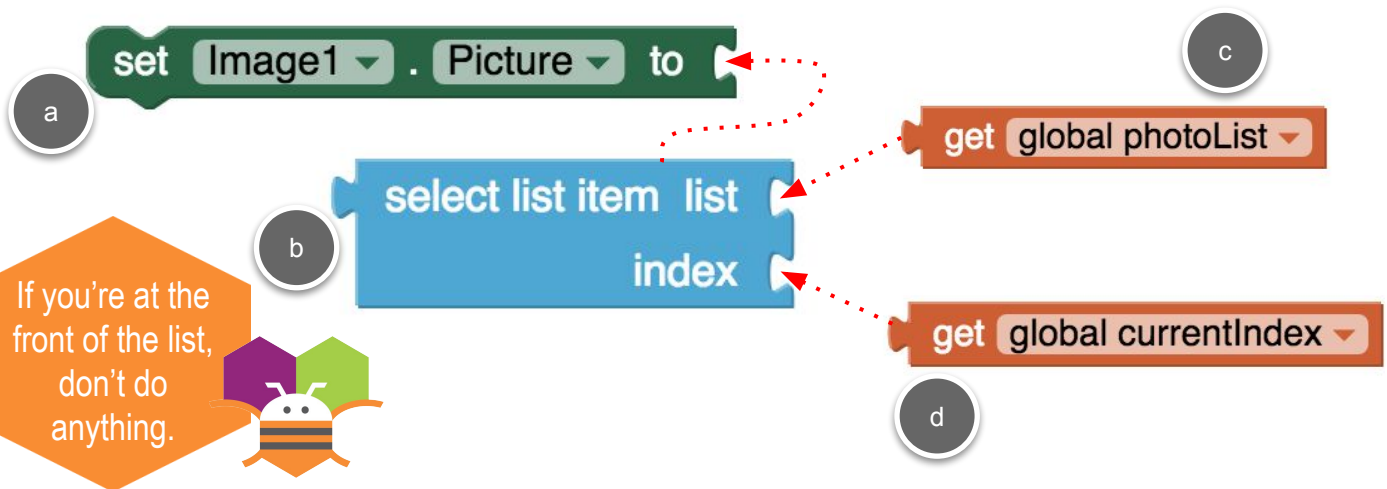
- 18 If you're not at the front of the list (index > 1)...



- 19 Then subtract 1 from the index.



- 20 And set the **Image1.Picture** to the photo at the new index.



NEXTBUTTON

- 21 Do you think you can figure out the **NextButton.Click** event? ----->



- 22 The **BackButton** works the same way it did in the **LocationScreen**. -----↓



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

- Go to the **GalleryScreen**.
- Take some pictures.
- Scroll through your pictures with the Previous and Next buttons.
- Close the app and reopen it.
The pictures should still be there!



COMPUTATIONAL THINKING CONCEPTS

The following are the Computational Thinking Concepts used in Multiple Pictures.

Tour Guide

1. Naming/Variables

initialize global currentIndex to 0 initialize global photoList to create empty list

2. Manipulation of data and elementary data structures

```

add items to list list
item
  get global photoList
  get image

when GalleryScreen.Initialize do
  set global currentIndex to 0
  set global photoList to
    call TinyDB1.GetValue
      tag "photoList"
      valueIfTagNotThere create empty list
  if not is list empty? list
    get global photoList
    then
      set global currentIndex to 1
      set Image1.Picture to
        select list item list
          index
          get global photoList
          get global currentIndex
  
```

3. Conditionals

```

if not is list empty? list
  get global photoList
  then
    set global currentIndex to 1
    set Image1.Picture to
      select list item list
        index
        get global photoList
        get global currentIndex
  
```

4. Operators

```

not is list empty? list
  get global photoList
  
```