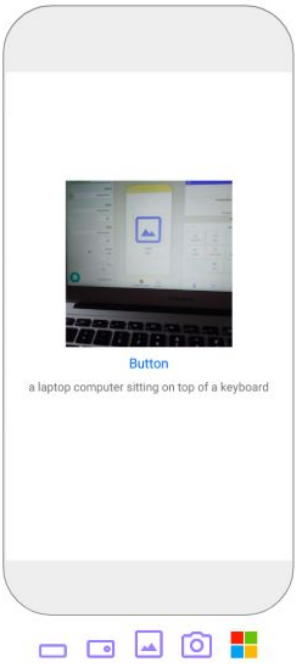
4. Image Recognition



Objective:
This app is a great example of how Thunkable can be used to create powerful, artificial intelligence apps with just a handful of blocks. By utilizing the built in ImageRecognizer, students can take a photo with the camera on their device and use cloud-based AI to come up with a suitable description for their picture. Since this app is slightly more complex than the previous apps students should be encouraged to incrementally, or iteratively, test their apps as they go.

Design Concepts:
Image [docs]
The image behaves a lot like a picture frame, you can place any image you like in the frame and you can even use code to change what is displayed in the image at any time.

Camera [docs]
Although you can see the camera controls and viewfinder on your screen the Camera is classed as an invisible component because in reality your Thunkable app is actually opening the native camera app on your device. When you have taken a picture the image is passed from the camera app back to your Thunkable app.

ImageRecognizer [docs]
Powered by Microsoft's "Cognitive Services", the ImageRecognizer component give your app computer vision without and registration or setup required.

Blocks:
Iterative testing
In this app we are introduced to the idea of building a little bit, testing, and building a bit more. This iterative approach for development should be highlighted and students should be encouraged to break larger problems into smaller, more manageable pieces as is the case in this app.

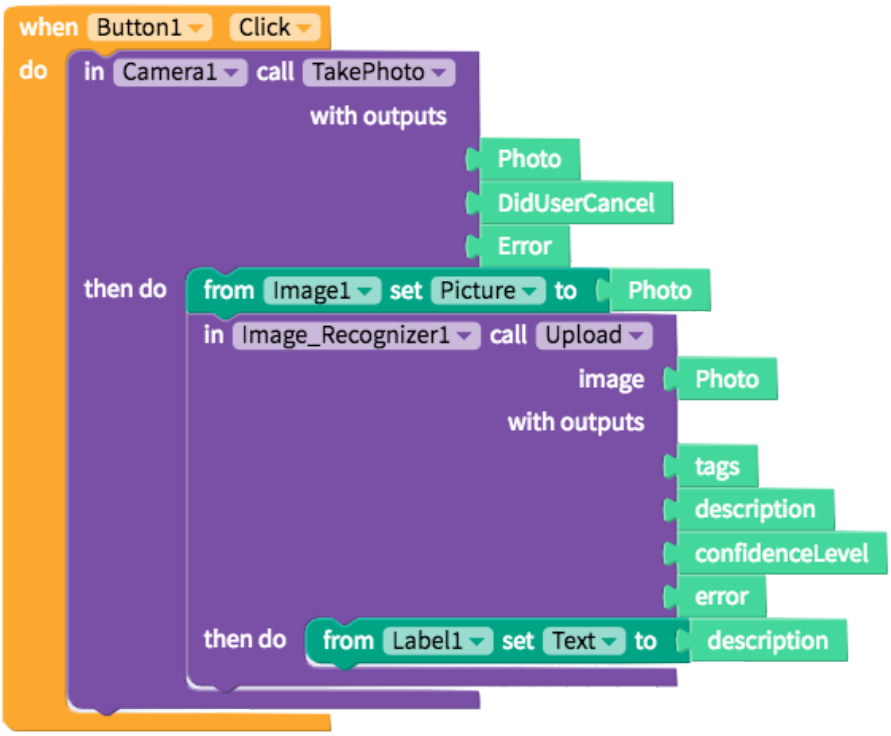
Call backs
We were introduced to asynchronous events in tutorial 2. In this tutorial a call back is used to handle what should happen when an asynchronous procedure eventually concludes.

Function parameters
This is the first occasion where we encounter procedures that have their own parameters.
Those of you coming from a mathematics background will already be familiar with the notion of passing arguments to a function and this is essentially the same thing.

Instructions:

1. Drag and drop a Button component into the phone.
2. From the Image section, drag and drop an Image component above the button.
3. From the Image section, drag and drop an ImageRecognizer component into the phone.
4. From the Image section, drag and drop a Camera component into the phone.
5. Click the blocks tab, then open the drawer for Button1. Drag and drop the "when Button1 Click" block into the blocks editor.
6. Open the drawer for Camera1. Select the "in Camera1 call TakePhoto" block, and drop it into the "when Button1 Click" block.
7. Open the drawer for Image1. Drag and drop the "from Image1 set Picture to" block under the "in Image_Recognizer1 call Upload" block. Next, drag another photo block from the Camera1 block, and drop it into the opening of the "from Image1 set Picture to" block.
8. Open the Thunkable Live App. Test to see if the image on the screen is set to the picture that you took with the camera.
9. Go back to the design section of the platform. In the User Interface section, drag and drop a Label component below the button.
10. In the Image section, drag and drop an Image Recognizer component into the phone.
11. Begin blocking out these components. Go to the Image_Recognizer1 drawer. Select the "in Image_Recognizer1 call Upload" block, and drop it into the "in Camera1 call TakePhoto" block.
12. Drag and drop the "Photo" block from the Camera1 block into the "image" opening on the Image_Recognizer1 block.
13. Open Label1's drawer. Drag and drop a "from Label1 set Text to" block inside the "in Image_Recognizer1 call Upload" block.
14. Drag and drop the "description" block from the Image_Recognizer1 block into the opening of the "from Label1 set Text to" block.
15. Congratulations! You just built your fourth app. Open up your Thunkable Live app and click Live Test on your computer to start using Microsoft Image Recognition.

Your final blocks should look like this:



Sample App:

Smart Camera http://bit.ly/thunk104

Progression:

- Use these blocks to add to your project! Try detecting whether or not a specific item was found in the picture. You do not need to use any additional components.