



Find It!

In this project you will make a mobile hide-and-seek game.

You will hide three objects around the room, and the player has to find them. Once they've found each item, they'll have to prove that they've found it by taking a photo of it.

Your mobile app will be able to analyse the photo to tell which object they've found.

You'll use machine learning to train your app to be able to recognise photos of the objects that you'll hide.

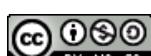
The screenshot shows the MIT App Inventor interface with the project titled "Find It!".

Blocks Editor:

- Blocks palette: labelSpace1, HorizontalArrangement1 (button_Start, button_Photo), labelSpace2, label_Status, Camera1, ML4K1, Clock1, Notifier1, Any component, Any Button, Any Camera, Any Clock, Any HorizontalArrangement, Any Label, Any ML4K, Any Notifier.
- Screen1 blocks:
 - when button_Photo .Click do call Camera1 .TakePicture
 - when Camera1 .AfterPicture do set label_Status .Text to image
 - when ML4K1 .GotClassification data classification confidence do set label_Status .Text to join get confidence "% confident that you found the " get classification
 - if get classification = Doggie then set label_Doggy .BackgroundColor to green
 - else if get classification = Monkey then set label_Monkey .BackgroundColor to green
 - else if get classification = Scissors then set label_Scissors .BackgroundColor to green
 - then set label_Doggy .BackgroundColor = green and set label_Monkey .BackgroundColor = green and set label_Scissors .BackgroundColor = green
 - then set Clock1 .TimerEnabled to false

Preview Screen:

- Display: 45 seconds
- Text: Doggie, Monkey, Scissors
- Buttons: Restart, Take photo
- Text at bottom: 91% confident that you found the Scissors
- Timer: 00:55 - 00:27



This project worksheet is licensed under a Creative Commons Attribution Non-Commercial Share-Alike License
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

1. Go to <https://machinelearningforkids.co.uk/> in a web browser

2. Click on “**Get started**”

3. Click on “**Log In**” and type in your username and password

If you can't remember your username or password, ask your teacher or group leader to reset it for you.

4. Click on “**Projects**” on the top menu bar

5. Click the “**+ Add a new project**” button.

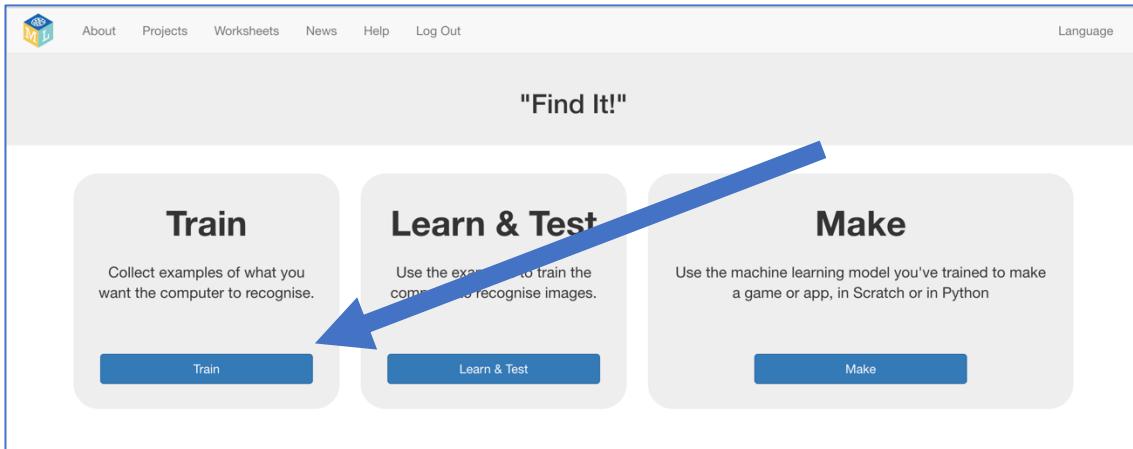
6. Name your project “Find It!” and set it to learn how to recognise “images”. Click **Create**

The screenshot shows a web page titled "Start a new machine learning project". At the top, there is a navigation bar with links for "About", "Projects", "Worksheets", "News", "Help", "Log Out", and "Language". Below the title, there is a "Project Name" field containing "Find It!". Underneath it, a "Recognising" dropdown menu is open, showing "images" as the selected option. To the right of the dropdown, a tooltip provides information about the choice: "What type of thing do you want the computer to recognise? For words, sentences or paragraphs, choose "text". For photos, diagrams and pictures, choose "images". For sets of numbers or multiple choices, choose "numbers.". At the bottom right of the form are two buttons: "CREATE" and "CANCEL".

7. You should see “Find It!” in the projects list. Click on it.

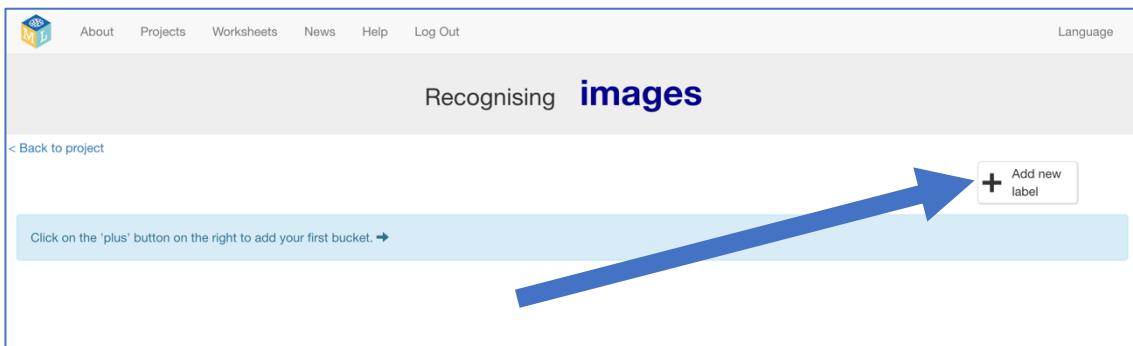
The screenshot shows a list of machine learning projects under the heading "Your machine learning projects". The first project in the list is "Find It!", which is described as "Recognising images". To the right of the project card is a "Delete" icon (a trash bin). Above the project cards, there is a button labeled "+ Add a new project".

8. We'll start by getting a collecting training examples. Click "Train"

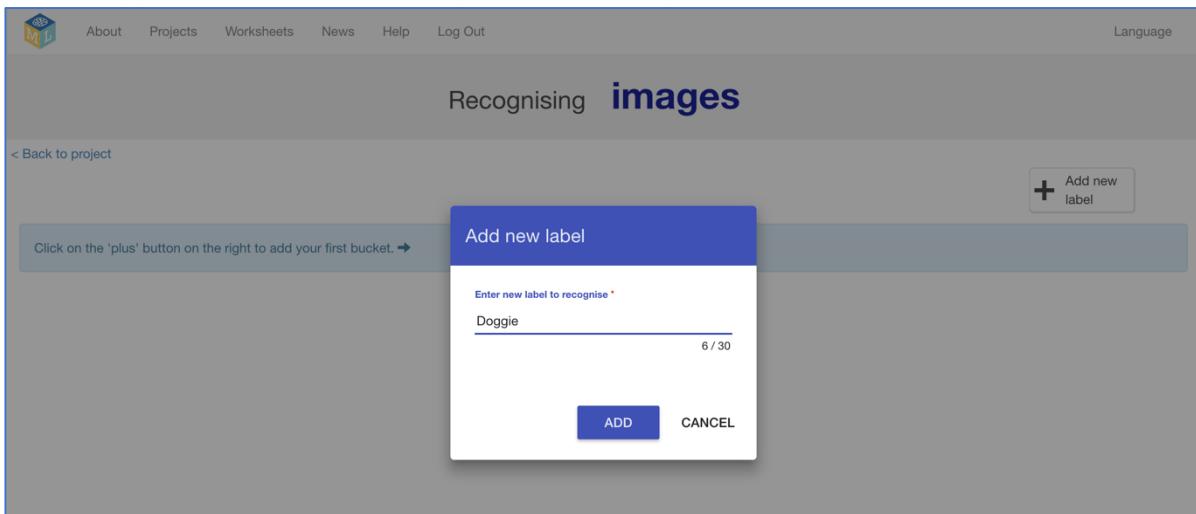


9. The game we're making is to challenge someone to find three objects that you will hide around the room. Choose your objects now. For the screenshots below, I'll be using a cuddly dog soft toy, a ceramic monkey, and a pair of scissors.

10. Click the "+ Add new label" button



11. Type in the name of your first object, and click "Add"



12. Repeat that for all three objects

Recognising **images** as **Doggie, Monkey or Scissors**

< Back to project

Doggie Drag pictures from other browser windows and drop them here

Monkey Drag pictures from other browser windows and drop them here

Scissors Drag pictures from other browser windows and drop them here

+ Add new label

www webcam draw

www webcam draw

www webcam draw

13. Click the “webcam” button in the first of your buckets

14. Take a photo of your first object using your computer webcam

Recognising **images** as **Doggie, Monkey or Scissors**

< Back to project

Doggie Drag pictures from other browser windows and drop them here

Add new example of Doggie

Scissors Drag pictures from other browser windows and drop them here

+ Add new label

www webcam draw

www webcam draw

www webcam draw

15. Repeat until you've got at least ten photos of your first object

Recognising **images** as **Doggie, Monkey or Scissors**

< Back to project

Doggie

Monkey Drag pictures from other browser windows and drop them here

Scissors Drag pictures from other browser windows and drop them here

+ Add new label

www webcam draw

www webcam draw

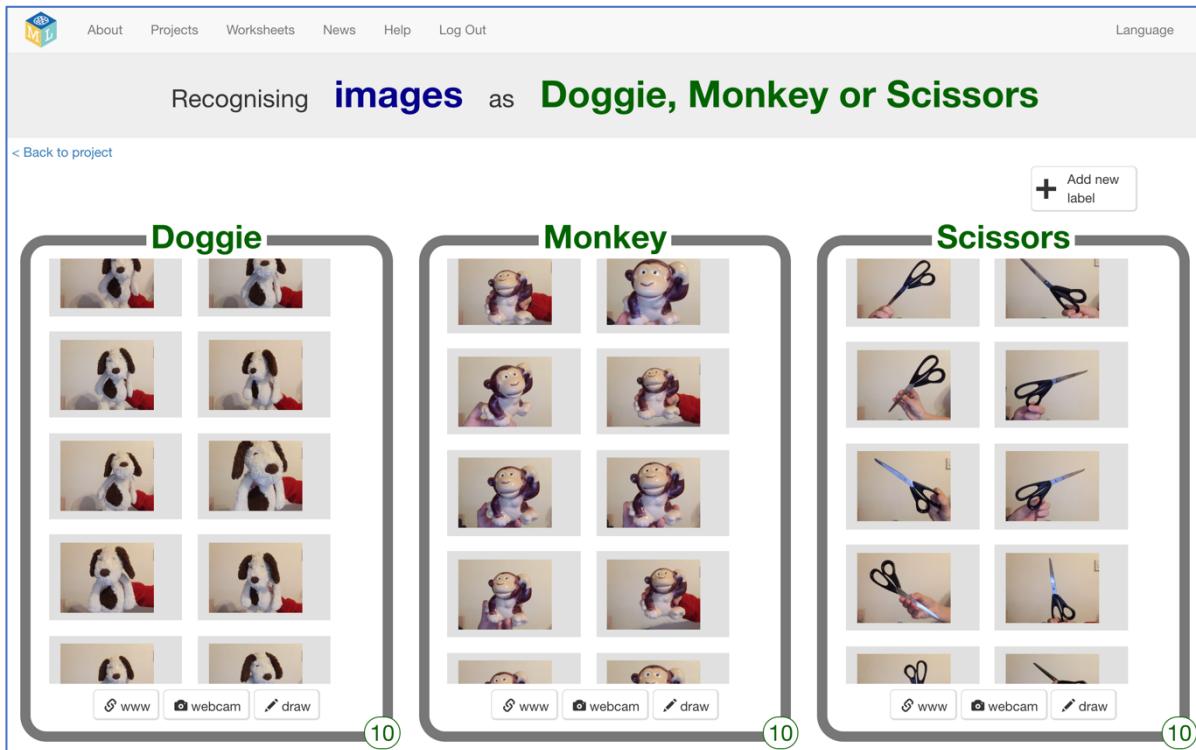
www webcam draw

10

16. Repeat that for all three objects.

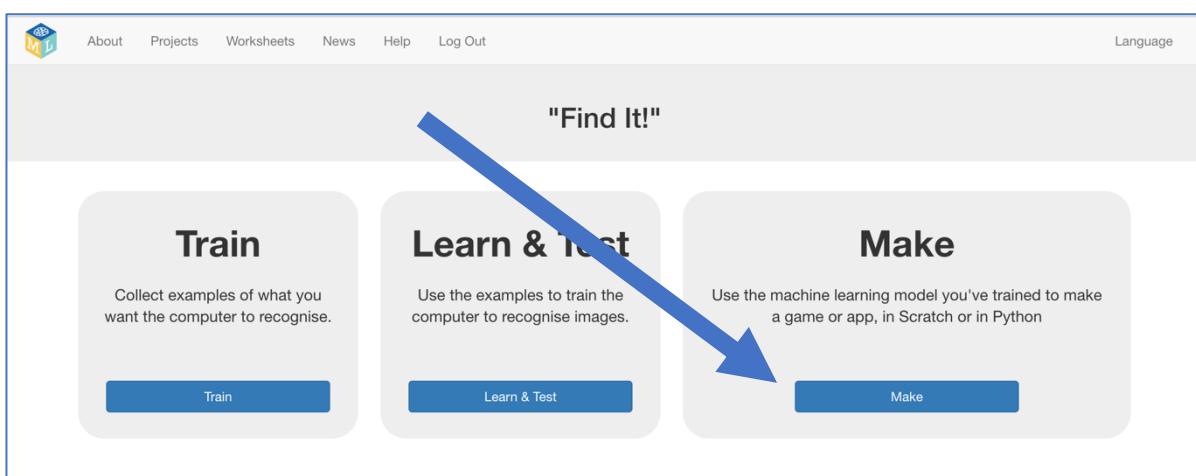
Use the webcam button at the bottom of each bucket to take a photo of that object. Take at least **ten** photos of **each** object.

Take a variety of photos of each object – from different sides and angles.



17. Click the “[< Back to project](#)” link

18. Finally, we'll use your machine learning model to make a game.
Click “[Make](#)”



19. Click on the “[App Inventor](#)” button

20. You will need the URL shown in red to set up your App Inventor project.

That is the unique address for your machine learning model.

The screenshot shows the App Inventor website with a blue arrow pointing from the text "To use App Inventor, go to <http://ai2.appinventor.mit.edu>" to a callout box containing a screenshot of the "Import an extension into project" dialog. The dialog has three numbered steps: 1. Click on Import extension, 2. Click on URL, and 3. Fill in the URL for your project. The URL field contains the redacted URL: <https://machinelearningforkids.co.uk/api/appinventor/b3c23199-eaea-11e8-8237-273d108bf05e26de1138-6a7c-226c-6a19-e5ca823d79c6/extension>.

21. Go to **App Inventor** at <http://ai2.appinventor.mit.edu>

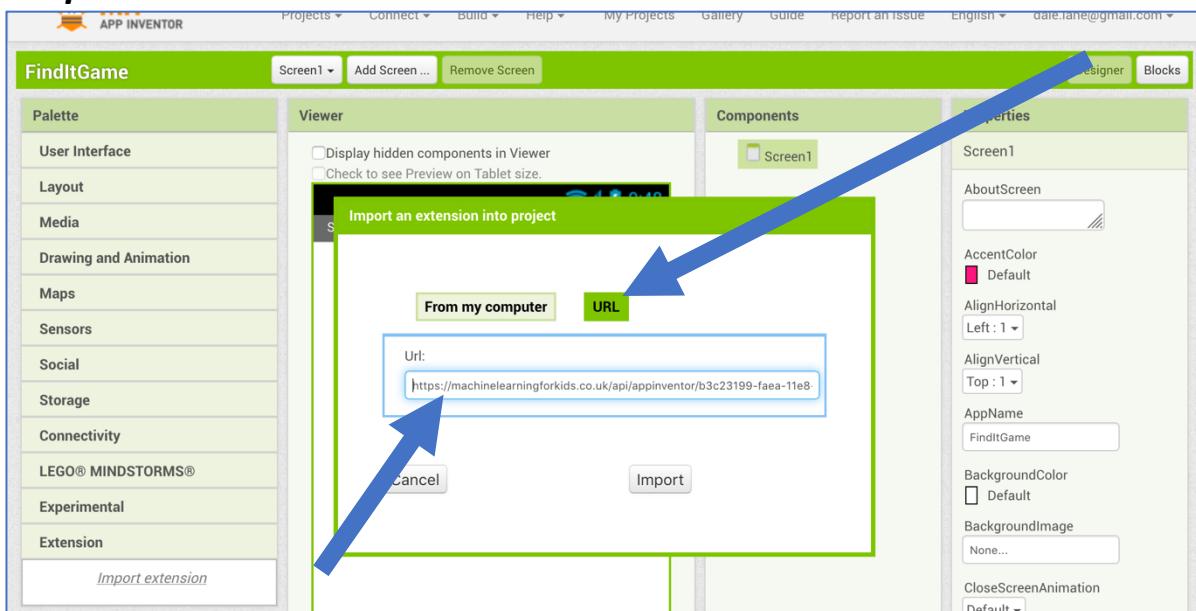
22. Start a new App Inventor project

23. Click on “Import Extension”

The screenshot shows the App Inventor Designer interface for the project "FindItGame". A blue arrow points from the "Import extension" button in the "Palette" section of the left sidebar to the "Import extension" dialog window. The dialog shows three steps: 1. Click on Import extension, 2. Click on URL, and 3. Fill in the URL for your project. The URL field contains the redacted URL: <https://machinelearningforkids.co.uk/api/appinventor/b3c23199-eaea-11e8-8237-273d108bf05e26de1138-6a7c-226c-6a19-e5ca823d79c6/extension>.

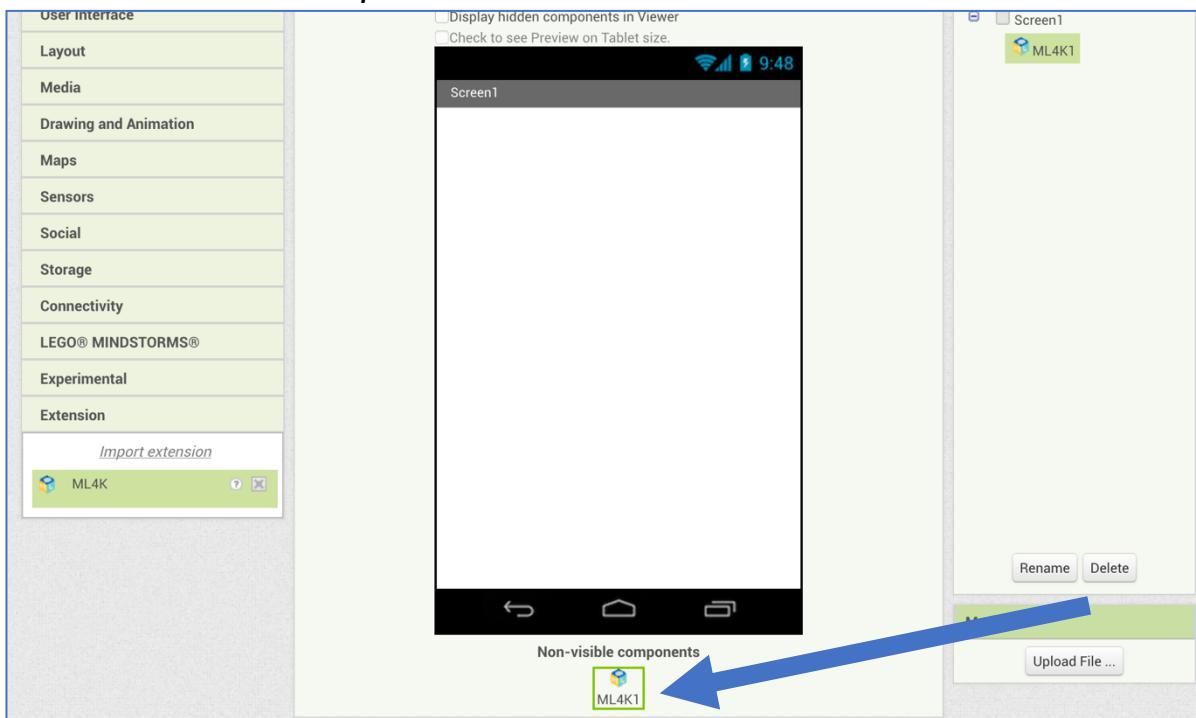
24. Import your machine learning model into the project.

Click on “URL”, then enter the URL that you got in step 23, and finally click “Import”.



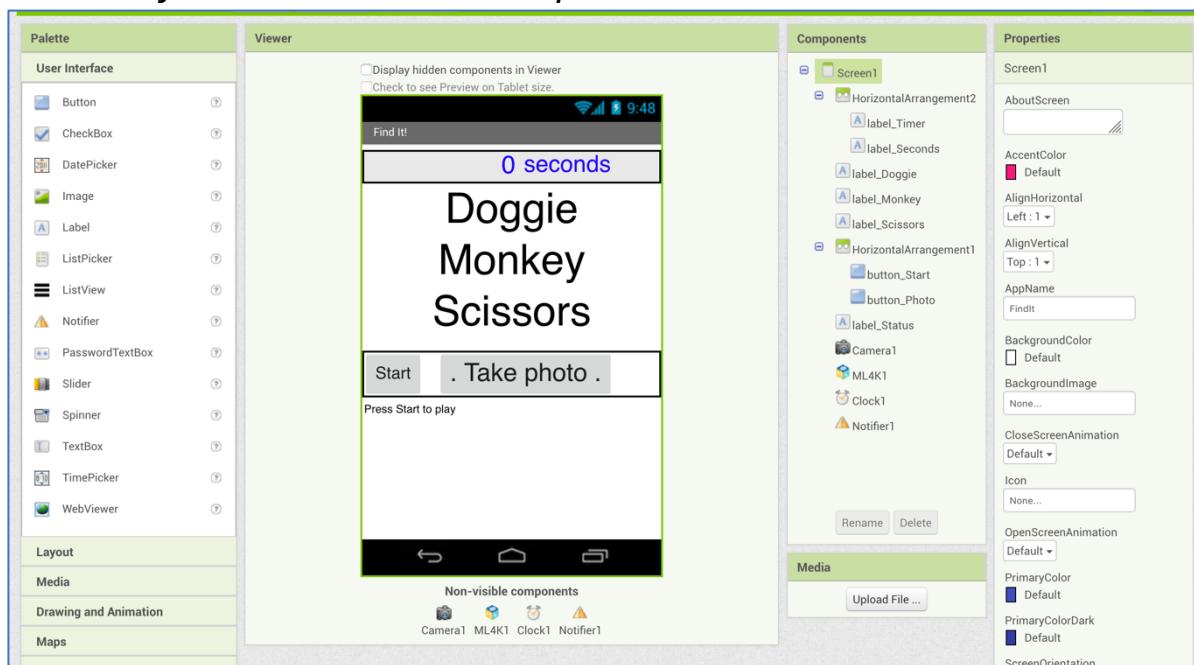
25. Drag the Machine Learning for Kids extension (“ML4K”) to the “Viewer”.

The icon will be added under the mobile screen once you’ve done this, in the “Non-visible components” list.



26. Create this mobile game user interface.

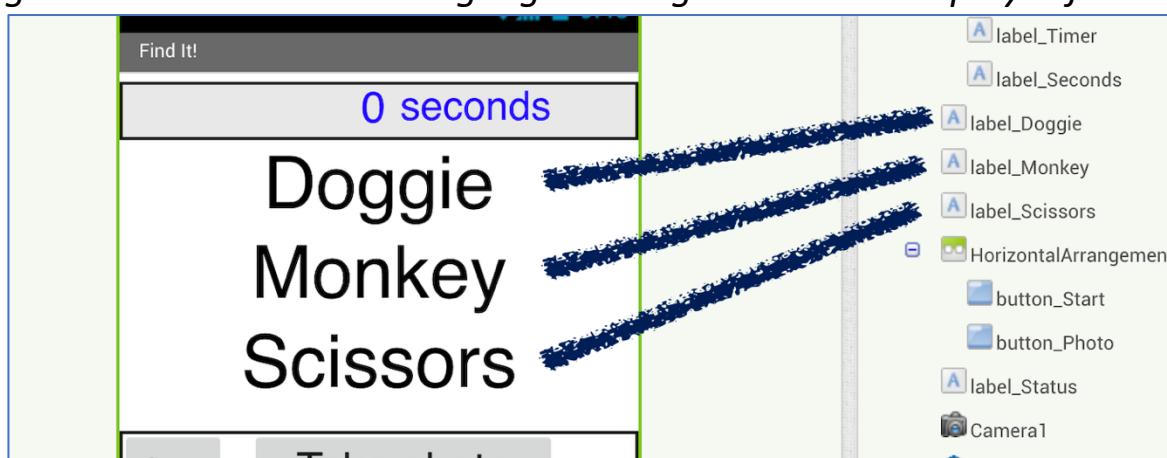
The next few screenshots will explain the main elements here.



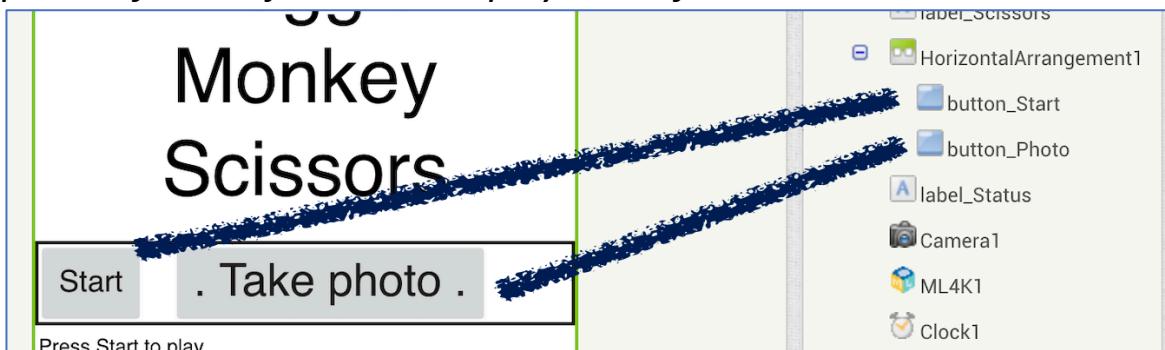
The game timer will keep track of how long it takes the player to find the objects. `label_Timer` will display the current time, and `label_Seconds` just says “seconds”.



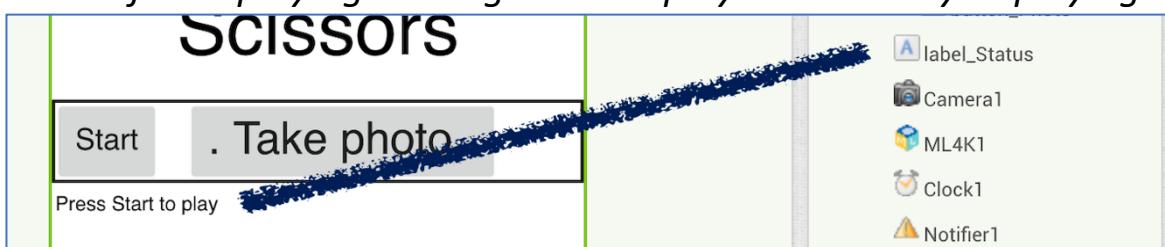
Three labels display the objects that you want the player to find in this game. Each item will be highlighted in green when the player finds them.



A couple of buttons to control the game – one to start, and one to take a photo of the object that the player has found.



A label for displaying messages to the player while they're playing.

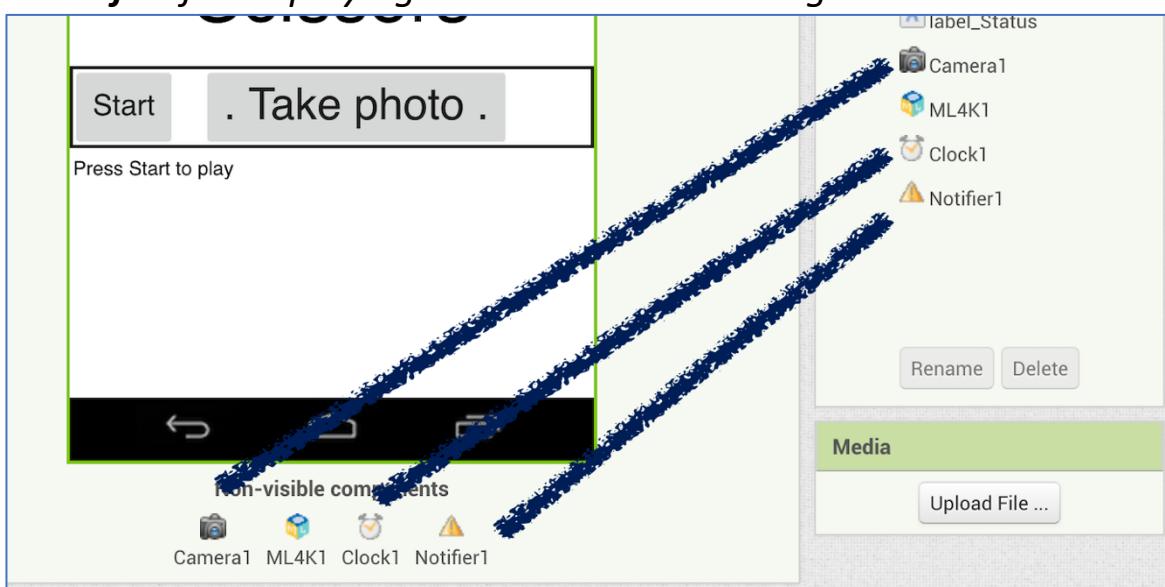


A collection of non-visible components needed in the game:

A Camera for taking pictures of objects.

*A Clock for the game timer. Set it to **disabled** so the timer won't start yet.*

A Notifier for displaying the Game Over message.



You can change the look of the app to make it your own, but you need to include these basic elements.

- 27.** Click on the “**Blocks**” button to start creating your script
- 28.** Create a script to start the game when the player clicks Start

```
when button_Start .Click
do
  set label_Doggie .BackgroundColor to [red v]
  set label_Monkey .BackgroundColor to [blue v]
  set label_Scissors .BackgroundColor to [green v]
  set label_Status .Text to "Try to find all the things!"
  set label_Timer .Text to "0"
  set Clock1 .TimerEnabled to [false v]
  set Clock1 .TimerEnabled to [true v]
  set button_Start .Text to "Restart"
  set button_Photo .Enabled to [true v]
```

- 29.** Create a script to display the game timer on the screen
-
- ```
when Clock1 .Timer
do
 set label_Timer .Text to [label_Timer .Text + (1)]
```

- 30.** Create a script to take a photo when the player clicks the button
- 
- ```
when button_Photo .Click
do
  call Camera1 .TakePicture
```

- 31.** Create a script to send photos to your machine learning model
-
- ```
when Camera1 .AfterPicture
image
do
 set label_Status .Text to "Thinking..."
 set button_Photo .Enabled to [false v]
 call ML4K1 .ClassifyImage
 path get [image v]
```

## 32. Create a script to display error messages if something goes wrong

```
when ML4K1 .GotError
 data error
 do set label_Status . Text to join " Something went wrong: " get error
 set button_Photo . Enabled to true
```

## 33. Create a script to update the game screen to confirm the object that they took a picture of.

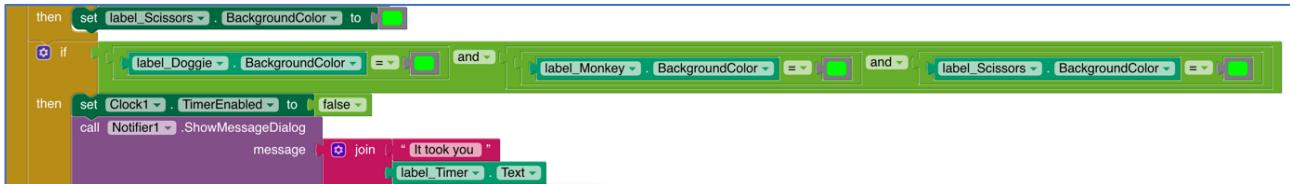
*The line that is cut off in the picture is shown on the next page.*

```
when ML4K1 .GotClassification
 data classification confidence
 do set label_Status . Text to join get confidence
 " % confident that you found the "
 get classification
 if get classification = " Doggie "
 then set label_Doggie . BackgroundColor to green
 else if get classification = " Monkey "
 then set label_Monkey . BackgroundColor to green
 else if get classification = " Scissors "
 then set label_Scissors . BackgroundColor to green
 if label_Doggie . BackgroundColor = and label_Monkey .
 then set Clock1 . TimerEnabled to false
 call Notifier1 . ShowMessageDialog
 message join " It took you "
 label_Timer . Text
 " seconds to find all the things. "
 title " You finished! "
 buttonText " OK "
 else set button_Photo . Enabled to true
```

**34.** The if line needs to check if every object is now green.

*If label\_A.BackgroundColor = Green and label\_B.BackgroundColor = Green  
and label\_C.BackgroundColor = Green*

*If every object is green, it means the player has found all three objects and the game ends.*



**35.** Add a script that trains a machine learning model on your Android device when your app starts

*This might take a little while, so the first time you start the app you will need to wait before pressing Start.*

*After the first time, your model will be reused so you won't need to wait.*



## What have you done?

You've trained a machine learning model to recognise photos of objects.

You did this by collecting examples of photos of these objects, and “labelling” them – telling the computer what is in each photo. The computer uses these labelled training examples to learn how to recognise new photos, by looking for patterns in the colours and shapes for examples.

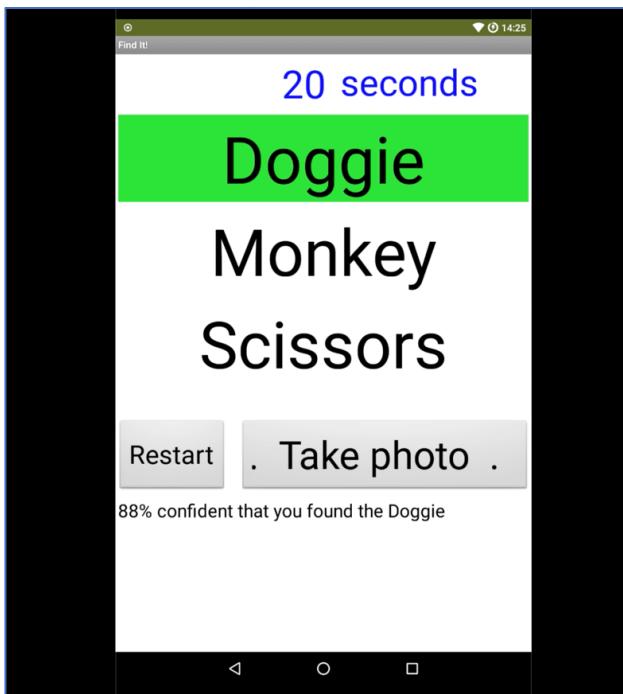
This is called “supervised learning” because of the way you had to tell the computer what each training example was, rather than leave it to learn for itself from a big group of photos of all three objects.

**36.** You're finished – test your game!

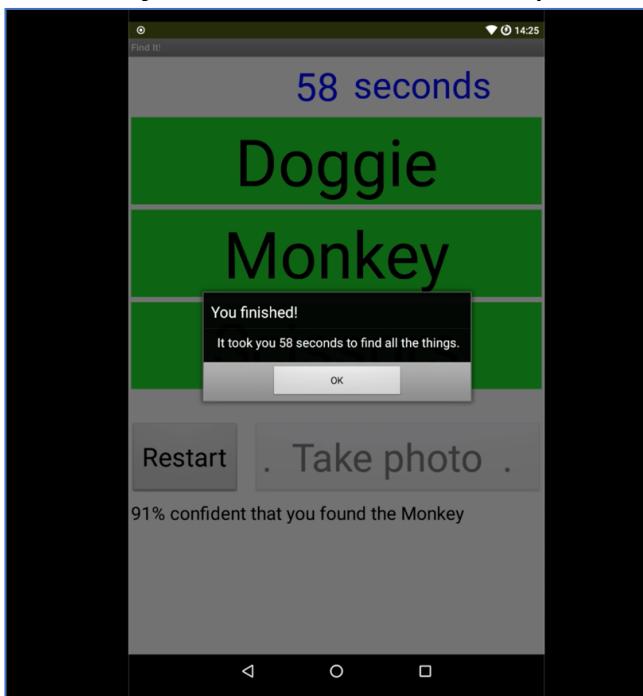
*Press Start then hunt for the three objects.*

*When you find an object, use the “Take photo” button to take a photo.*

*If your machine learning model verifies that you've found the object, it will be marked on your list.*



*When your machine learning model has verified that you've found all three objects, the timer will stop and report how long you took.*



## Tips, ideas and Extensions

Now that you've finished, why not give one of these ideas a try?

### Background

If the background of photos taken while playing the game are very different from the background of photos you took with the webcam to collect training examples, you might find that your machine learning model gets confused between objects and makes mistakes.

If that happens, try adding training examples with different backgrounds, to train the machine learning model to cope with a variety of backgrounds.

### Mix things up with your examples

Take photos of the objects close to the camera and far away. Take photos of the objects from every side, upside down, from the top and from the bottom.

The more variety you can get in your training examples, the better your machine learning model will perform.

### Try using confidence limits

The App Inventor block will return the confidence score for how certain your machine learning model is that it has recognised the photo.

What should your game do if the confidence score is very low?

Can you modify the script to use the score?