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.

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**"

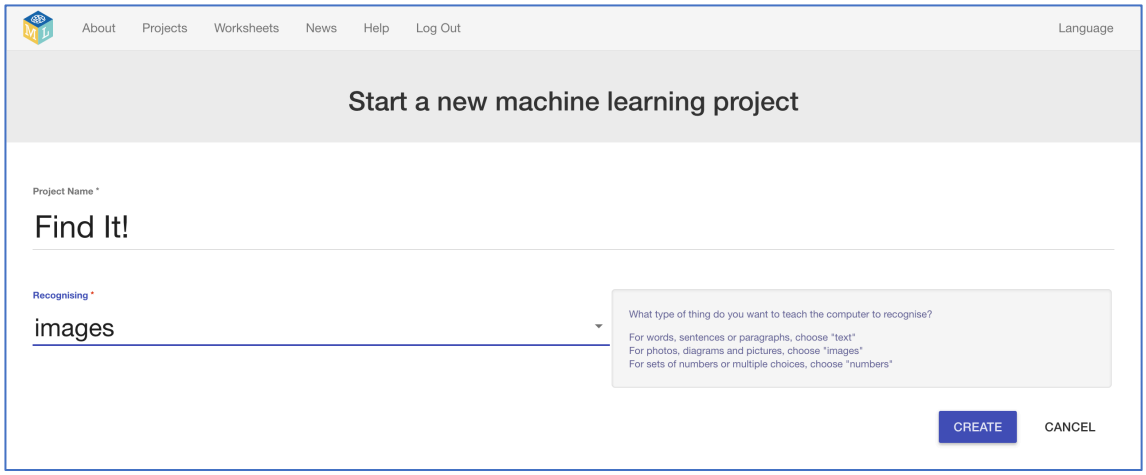
3. Click on "**Log In**" and type in your username and password

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

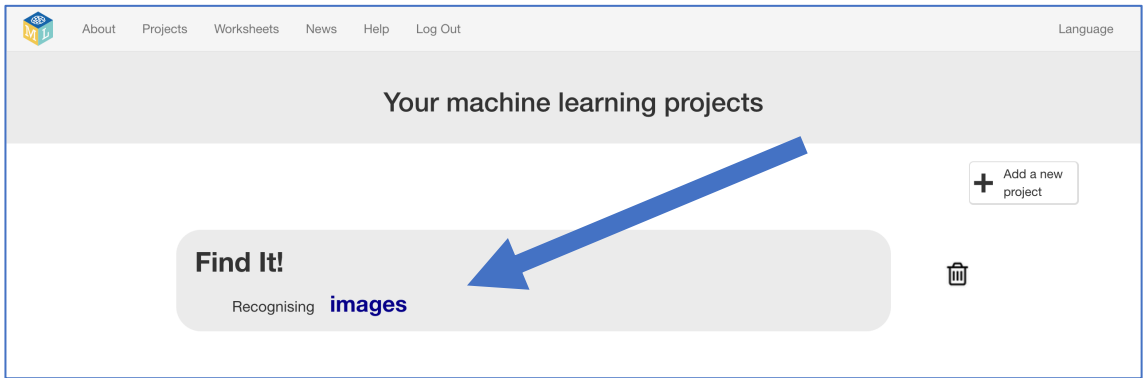
4. Click on "**Projects**" on the top menu bar

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**



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

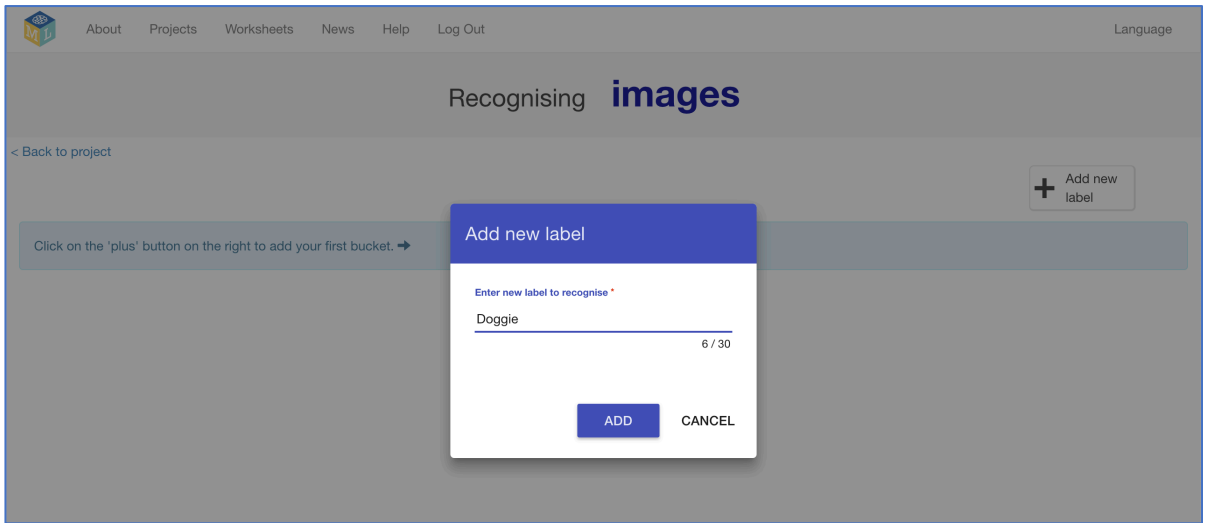


8. We'll start by getting a collecting training examples. Click "**Train**"

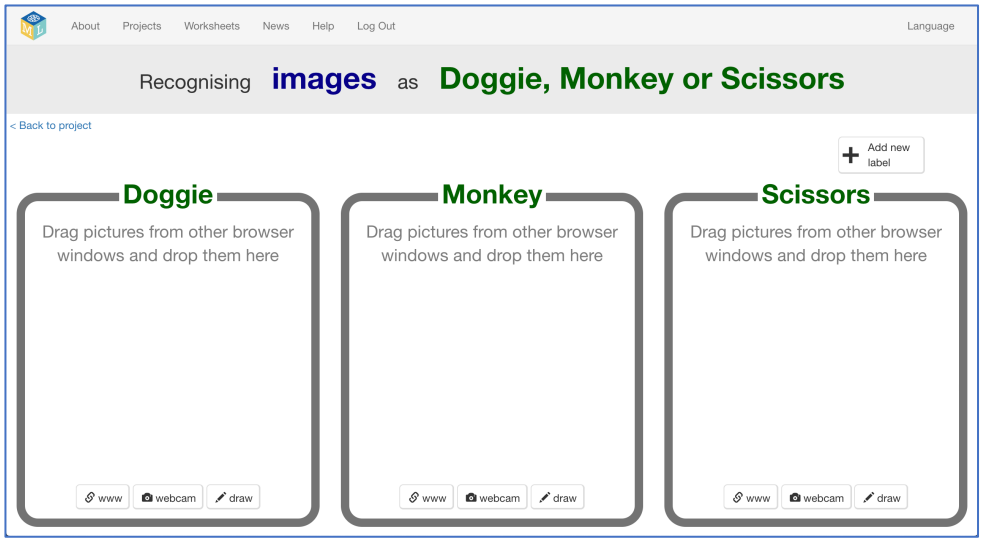
9. The game we're making is to challenge someone to find three objects that you will hide around the room. Choose your objects now. For the screenshots below, I'll be using a cuddly dog soft toy, a ceramic monkey, and a pair of scissors.

10. Click the "**+ Add new label**" button

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

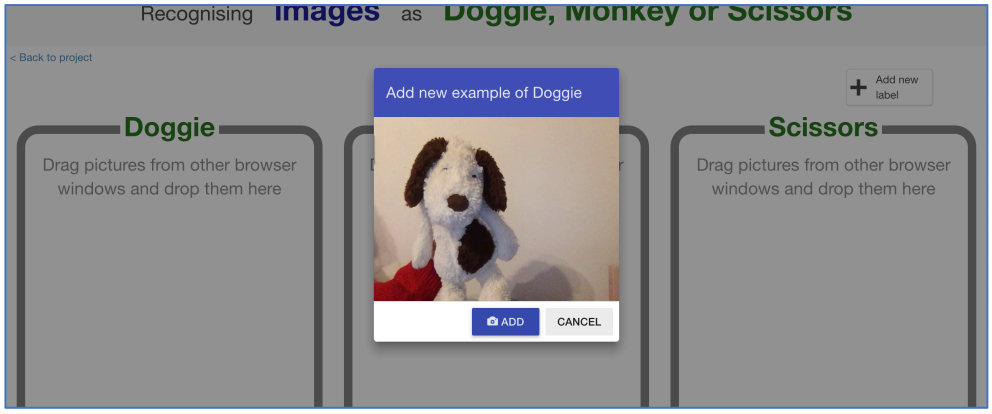


12. Repeat that for all three objects

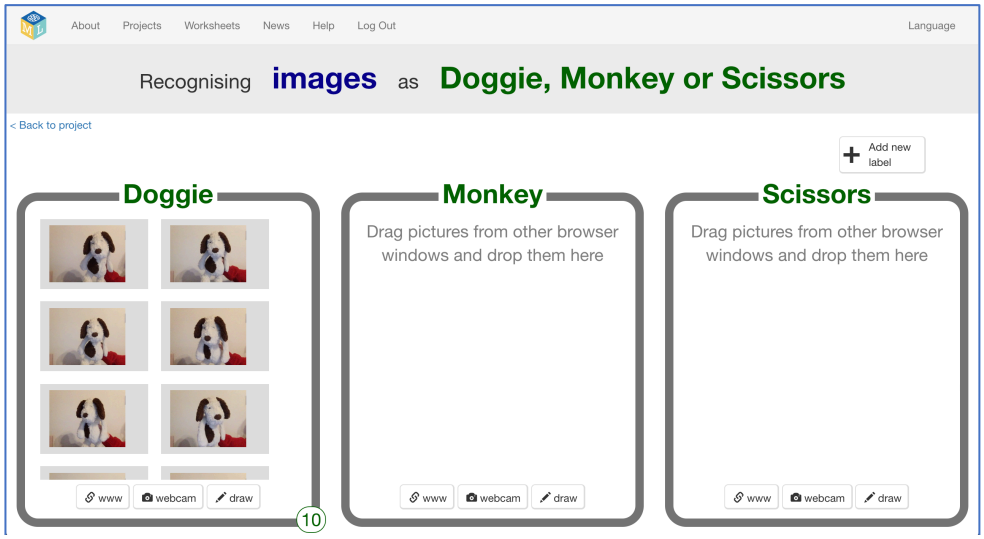


13. Click the "**webcam**" button in the first of your buckets

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



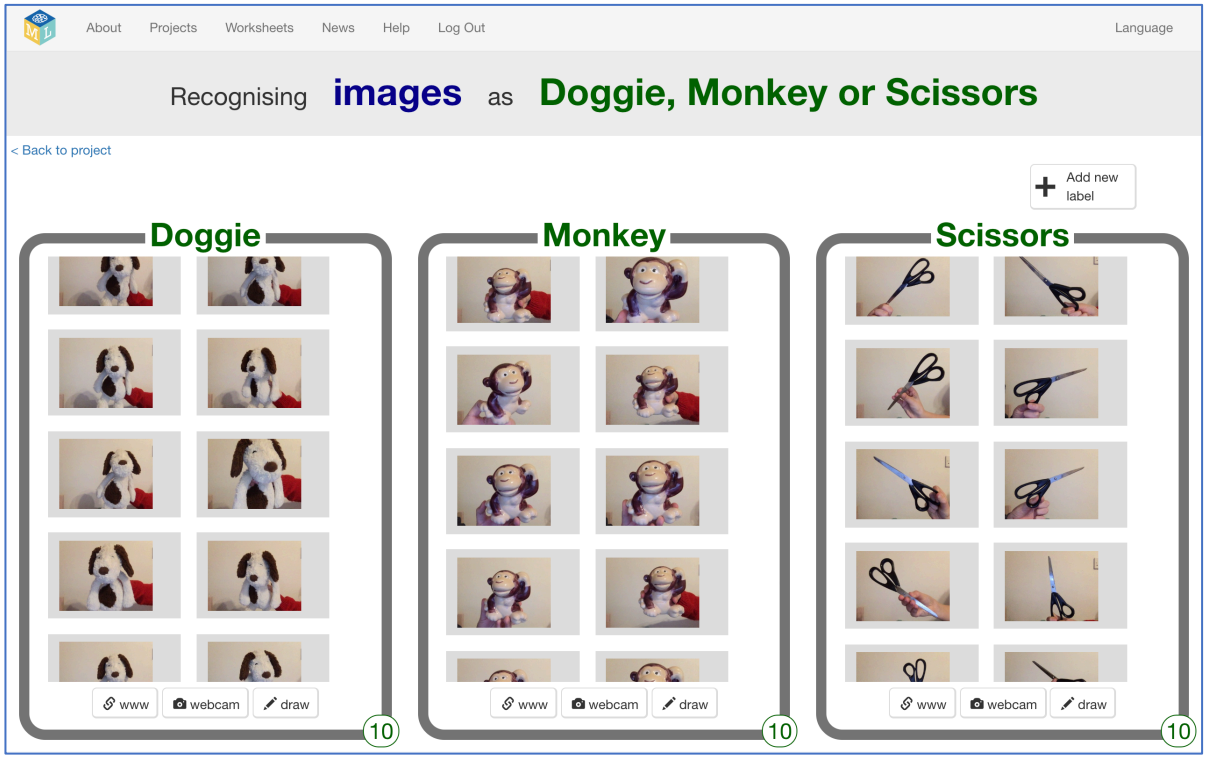
15. Repeat until you've got at least **ten** photos of your first object



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.

Try to take a variety of photos of each object – from different sides and angles.



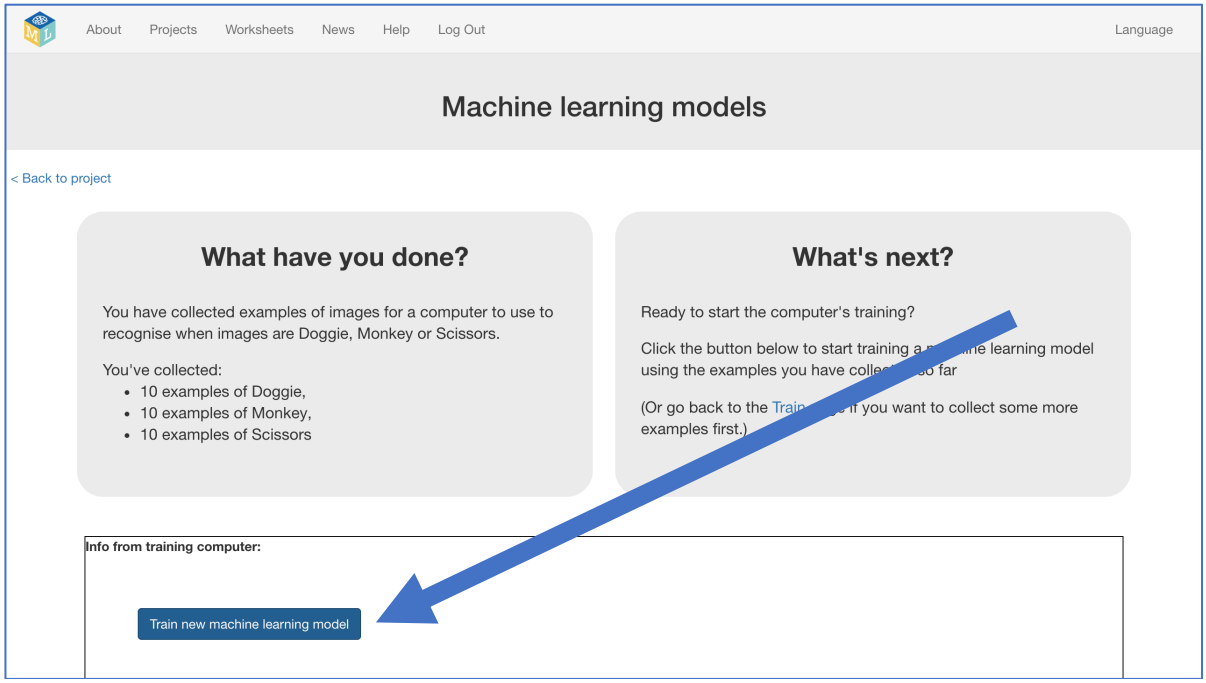
17. Click the "**< Back to project**" link

18. Next, we'll use your examples to train a machine learning model.

Click "**Learn & Test**"

19. Click the "**Train new machine learning model**" button

This will take a few minutes, but you can carry on and start building your mobile app while this is happening.

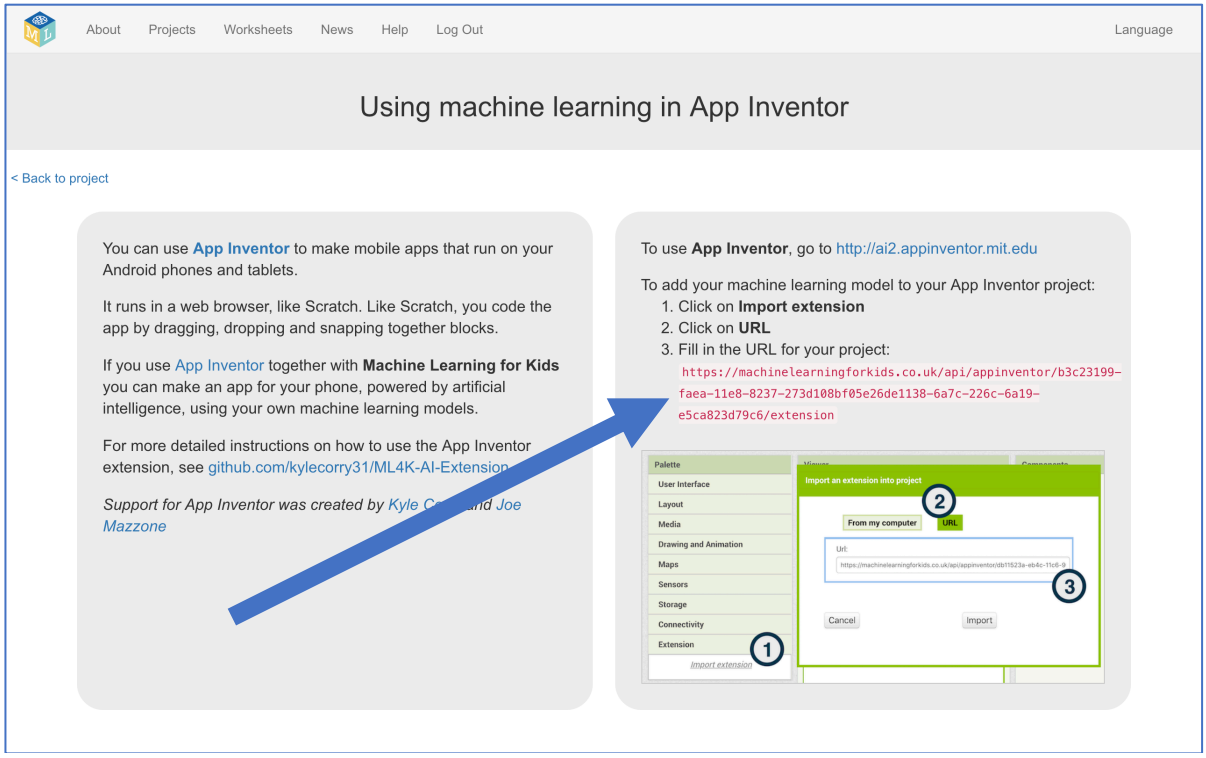


20. Click the "**< Back to project**" link

21. Finally, we'll use your machine learning model to make a game. Click "**Make**"

22. Click on the "**App Inventor**" button

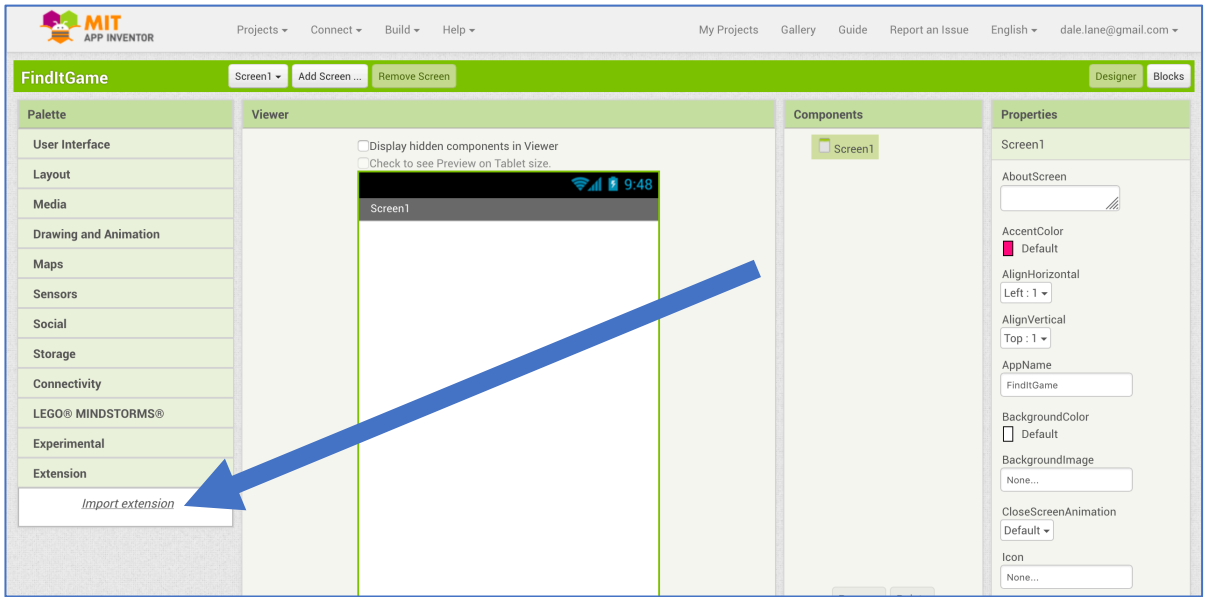
23. You will need the URL shown in red to set up your App Inventor project.
That is the unique address for your machine learning model.



24. Go to **App Inventor** at http://ai2.appinventor.mit.edu

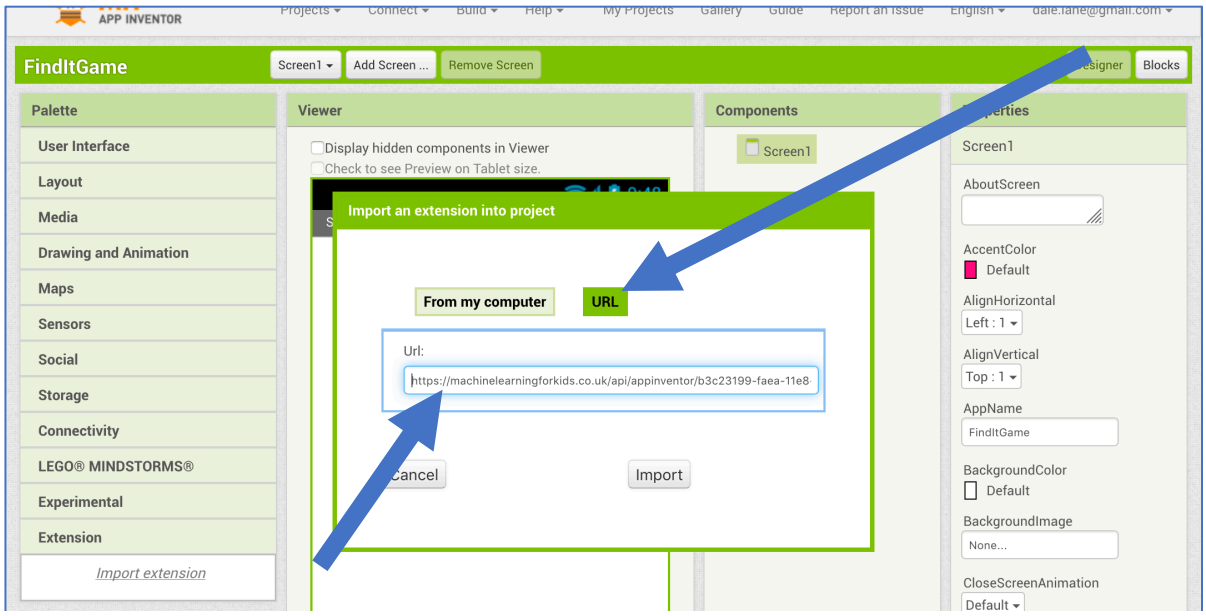
25. Start a new App Inventor project

26. Click on "**Import Extension**"



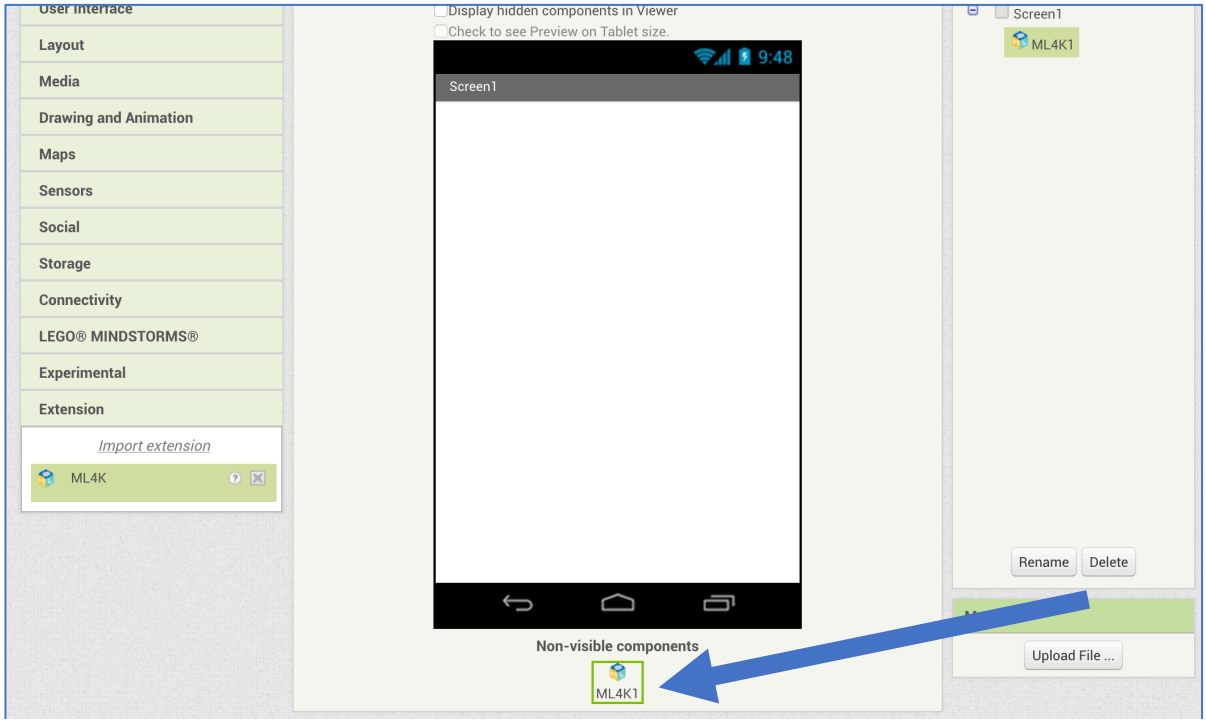
27. Import your machine learning model into the project.

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



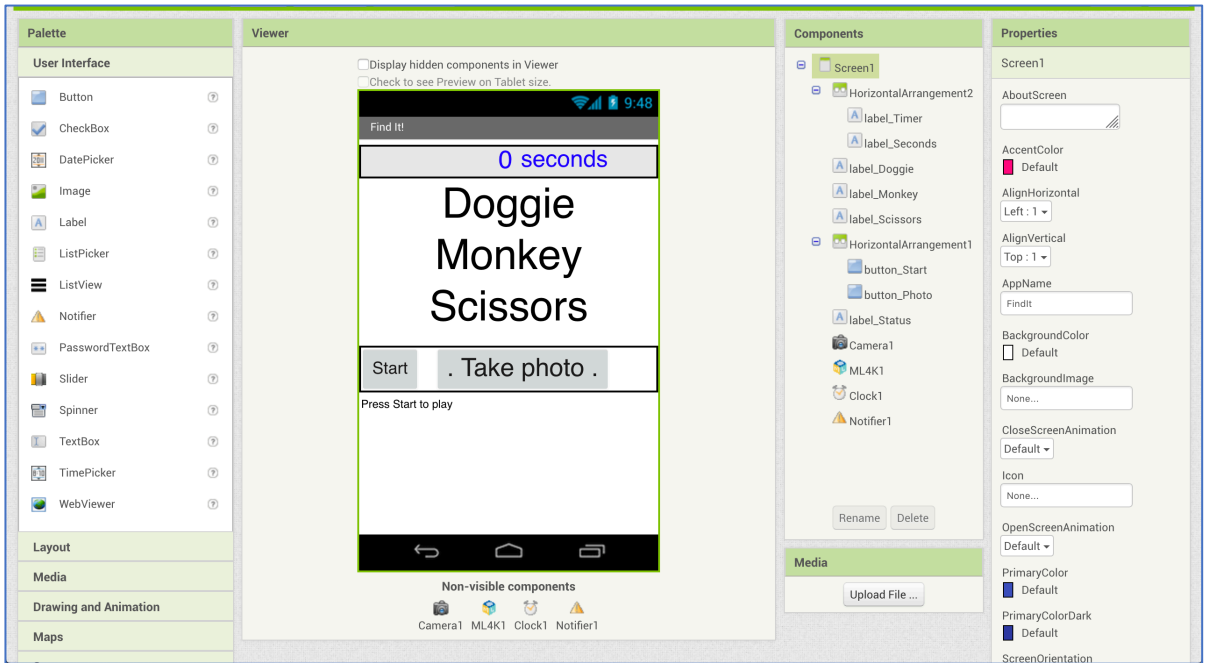
28. Drag the Machine Learning for Kids extension ("**ML4K**") to the "**Viewer**".

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



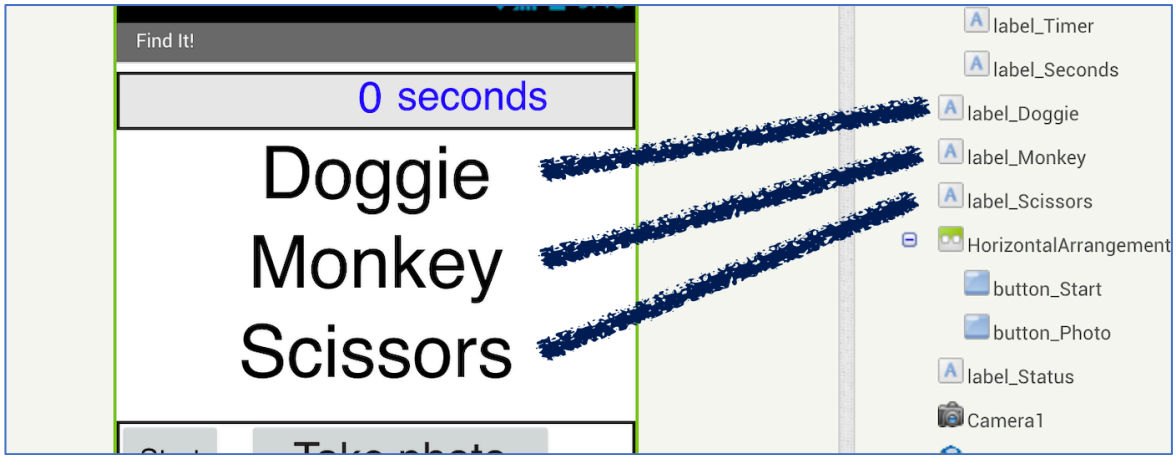
29. Create this mobile game user interface.

The next few screenshots will explain the main elements here.

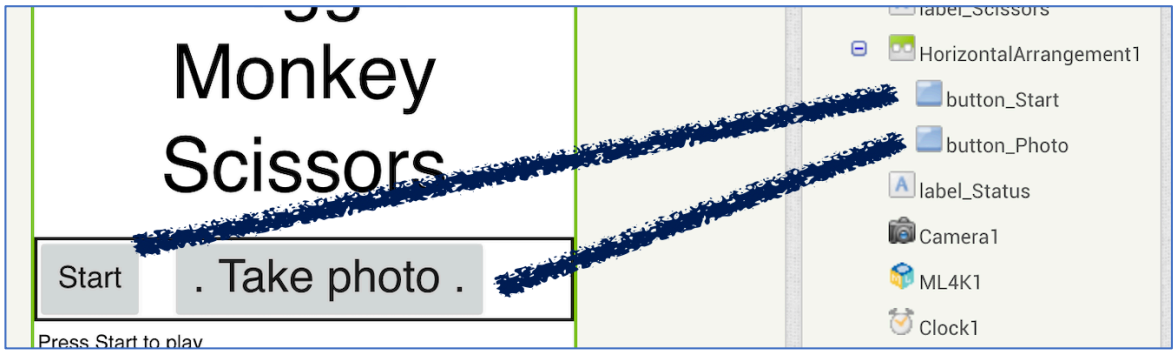


The game timer will keep track of how long it takes the player to find the objects. label_Timer will display the current time, and label_Seconds just says "seconds".

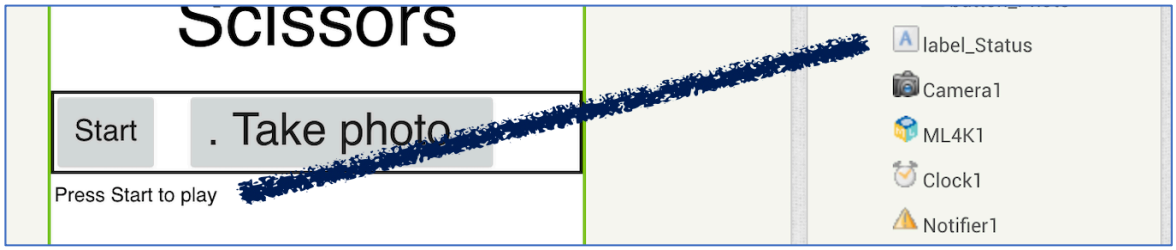
Three labels display the objects that you want the player to find in this game. Each item will be highlighted in green when the player finds them.



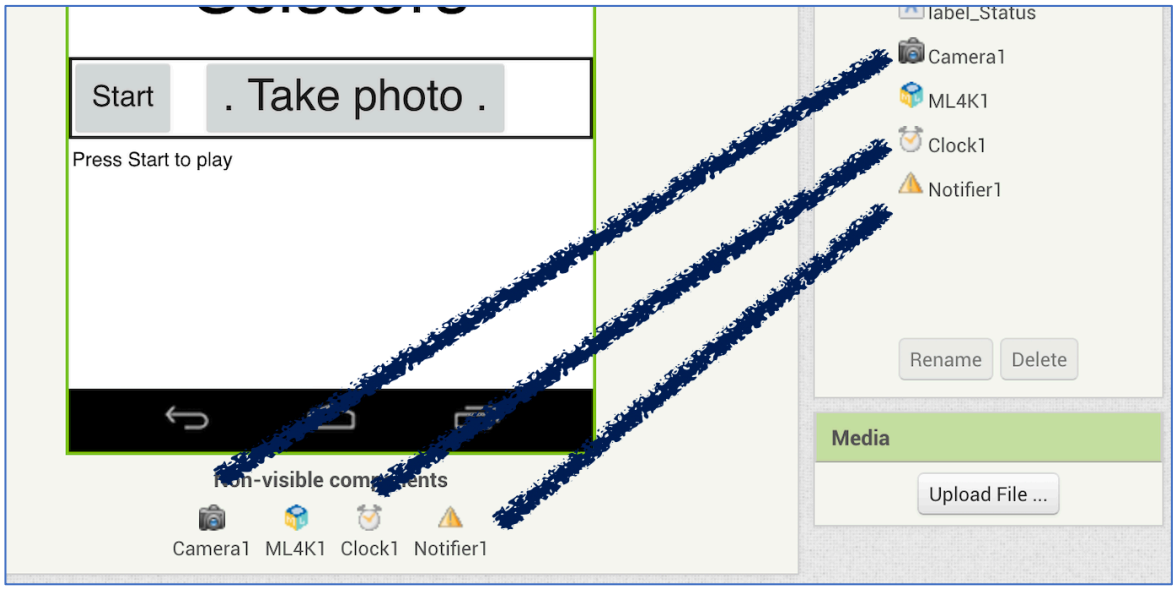
A couple of buttons to control the game – one to start, and one to take a photo of the object that the player has found.



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



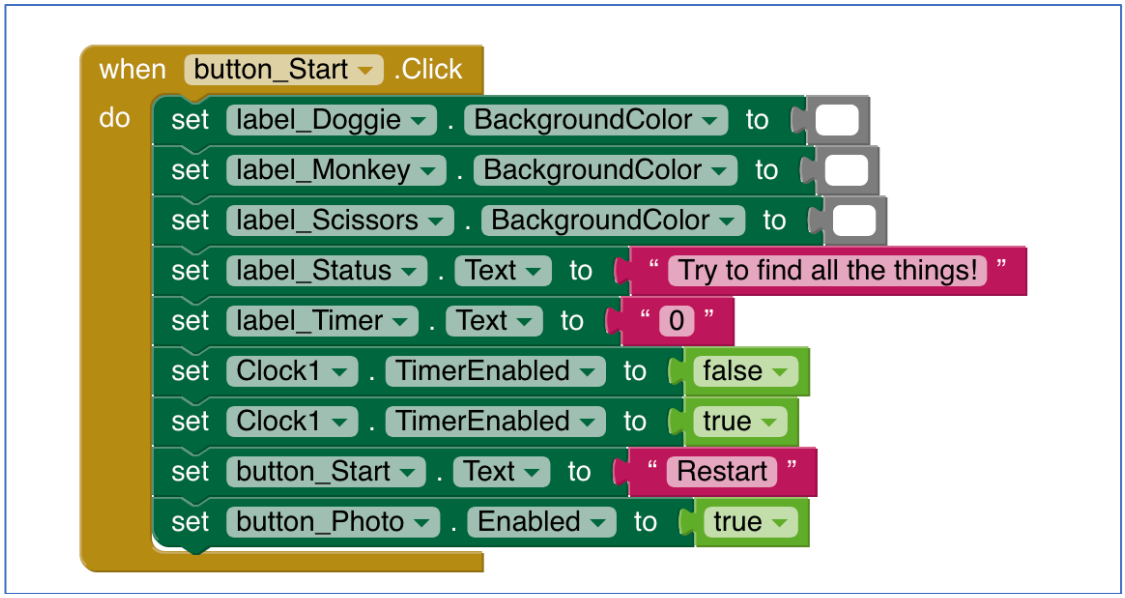
A collection of non-visible components needed in the game:
*A **Camera** for taking pictures of objects.*
*A **Clock** for the game timer. Set it to **disabled** so the timer won't start yet.*
*A **Notifier** for displaying the Game Over message.*



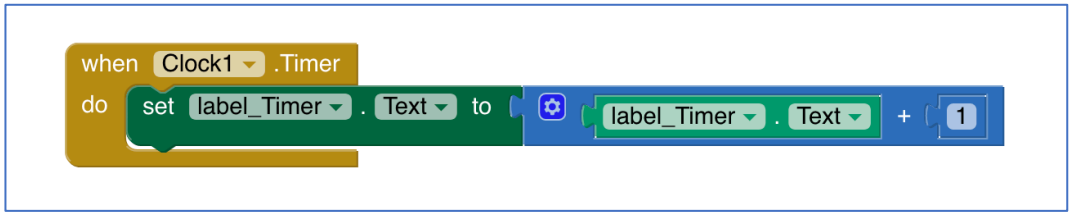
You can change the look of the app to make it your own, but you need to include these basic elements.

30. Click on the "**Blocks**" button to start creating your script

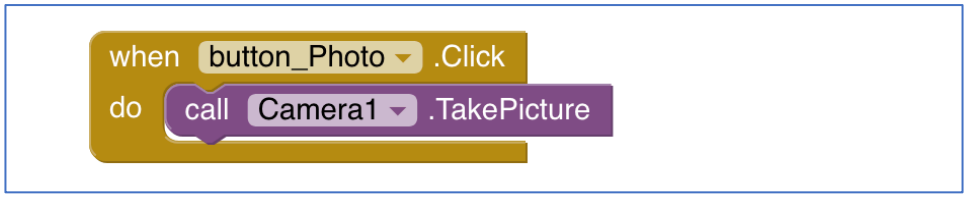
31. Create a script to start the game when the player clicks Start



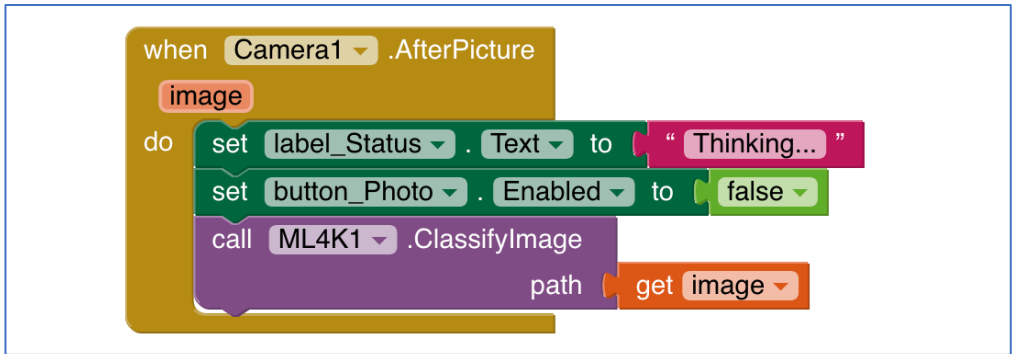
32. Create a script to display the game timer on the screen



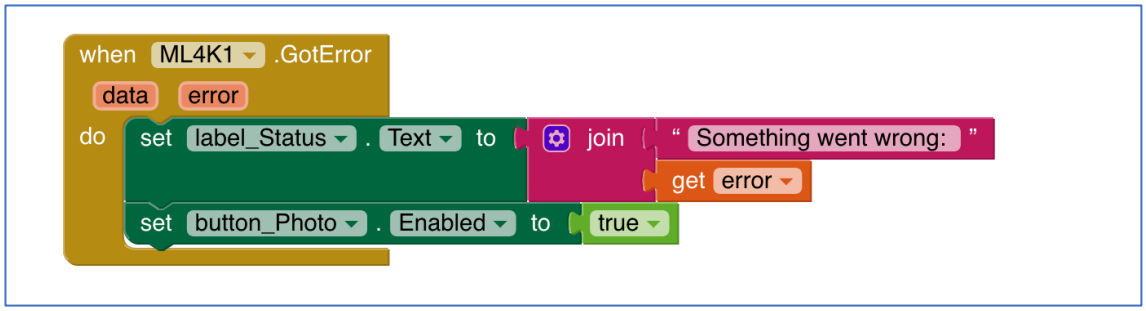
33. Create a script to take a photo when the player clicks the button



34. Create a script to send photos to your machine learning model

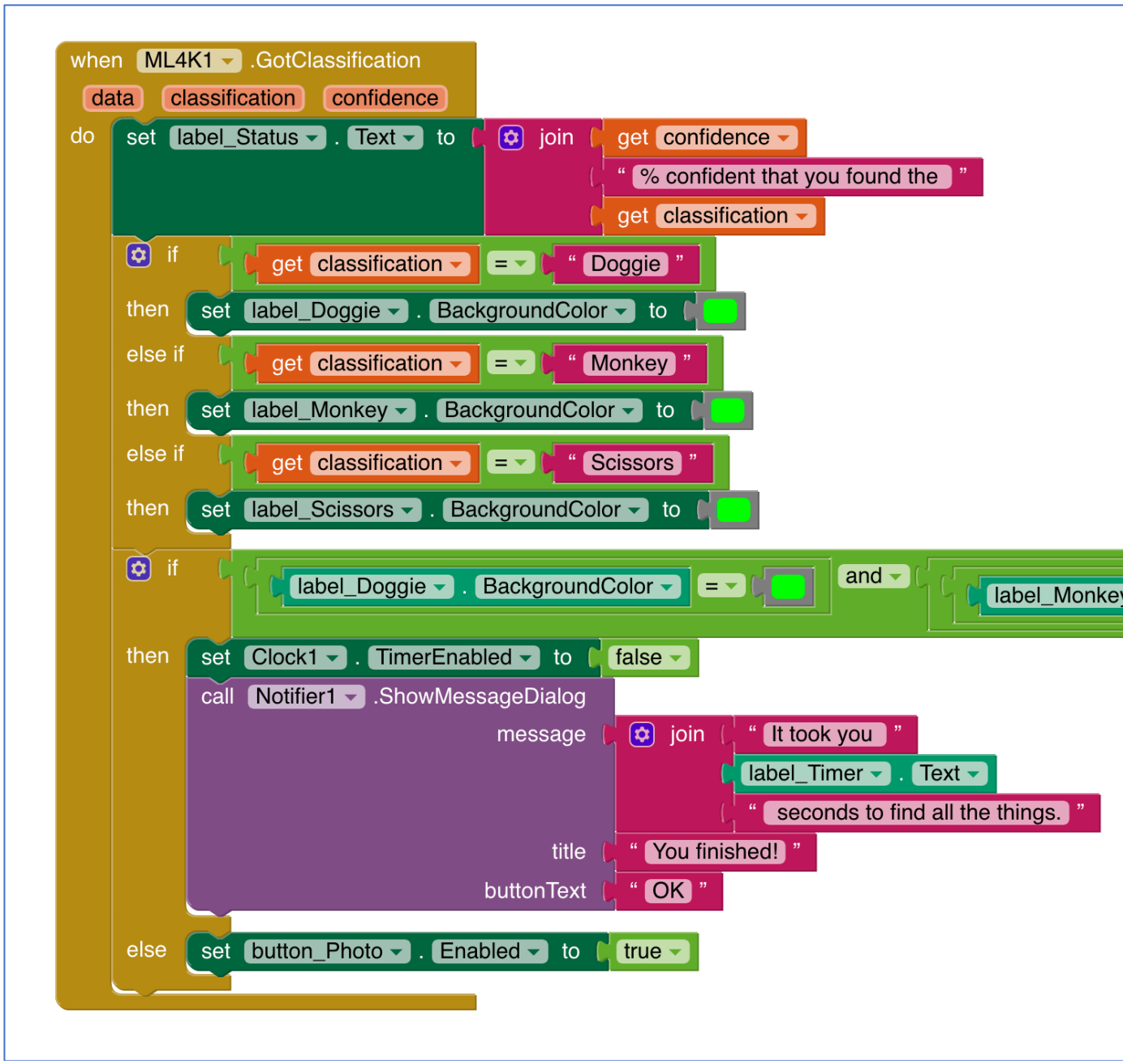


35. Create a script to display error messages if something goes wrong



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

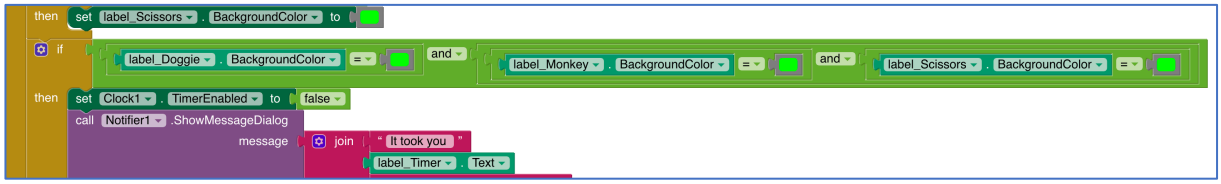
The line that is cut off in the picture is shown on the next page.



37. 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.

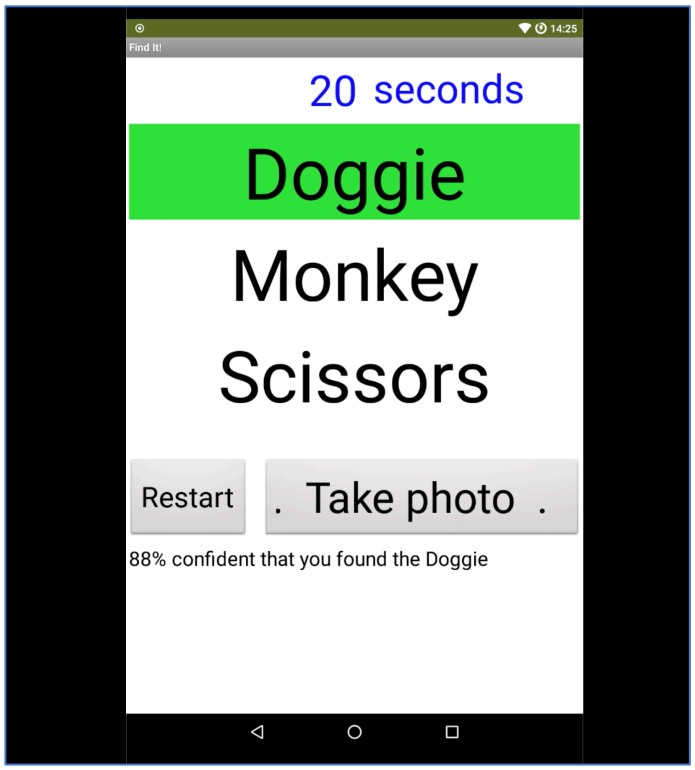


38. You're finished – test your game!

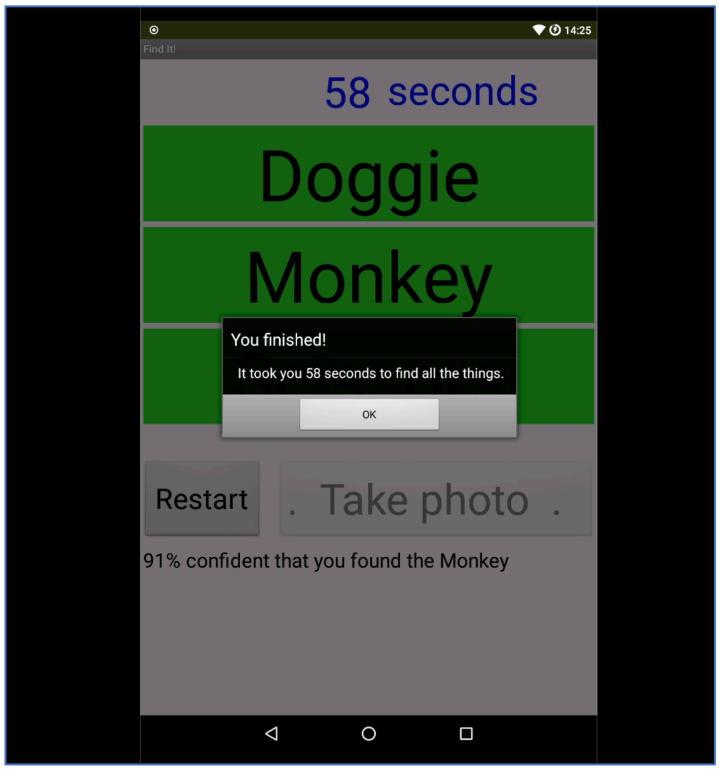
Press Start then hunt for the three objects.

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.



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.

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?