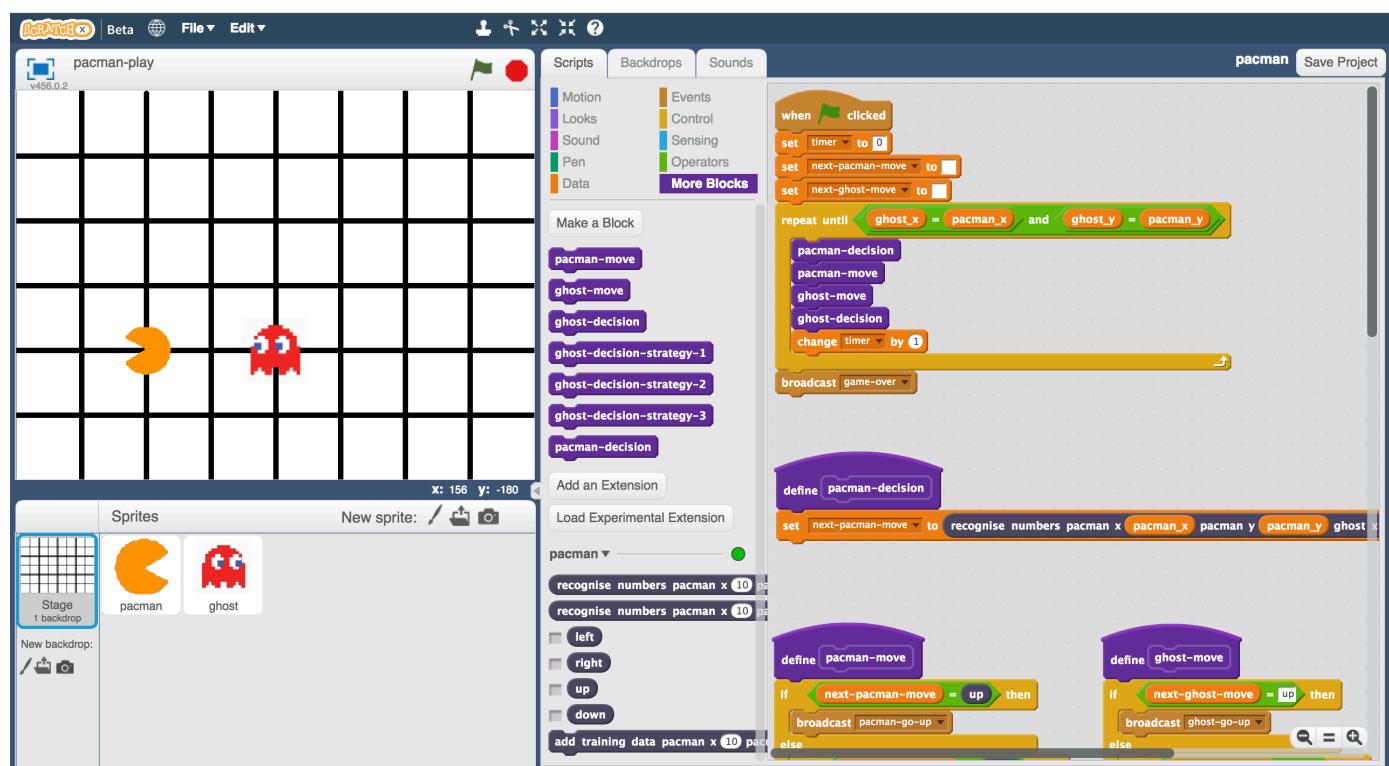


# Pac-Man

In this project you will create a Pac-Man game in Scratch that is able to learn from how you play.

You won't give it instructions for how to play, or tell it what the objective or rules of the game are.

Instead, you'll show it examples of you playing the game.

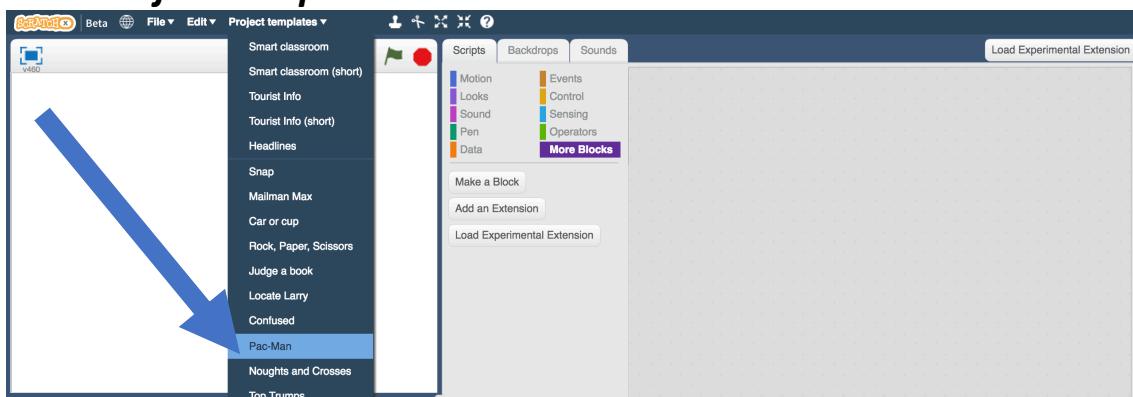


This project worksheet is licensed under a Creative Commons Attribution Non-Commercial Share-Alike License  
<http://creativecommons.org/licenses/by-nc-sa/4.0/>

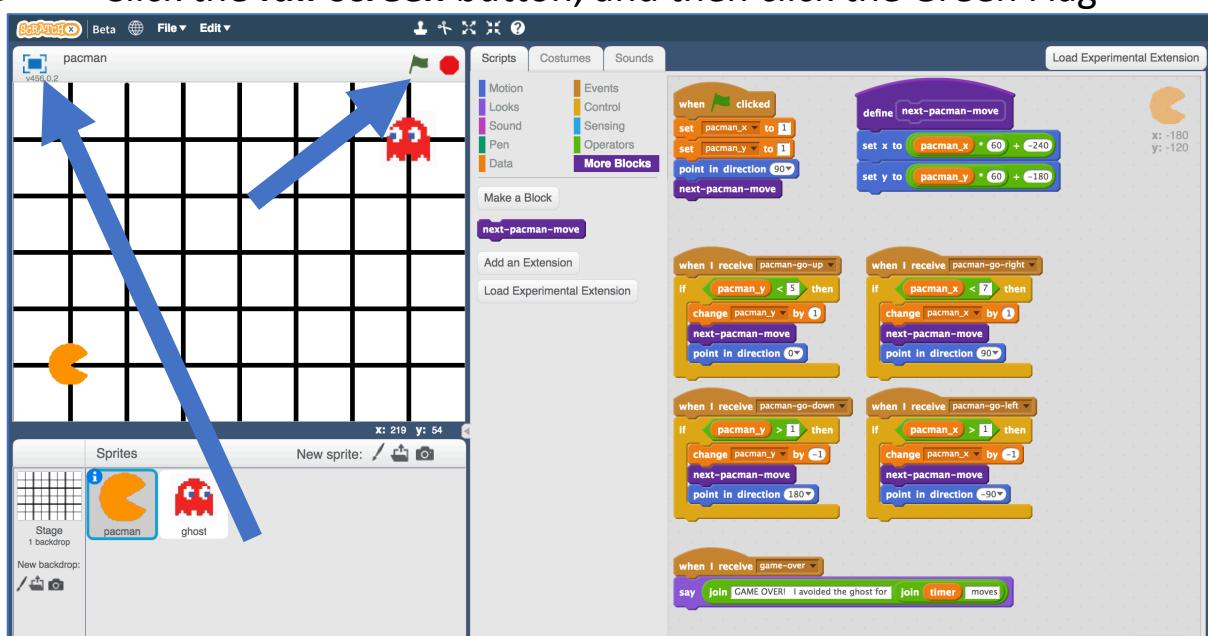
1. Go to <https://machinelearningforkids.co.uk/scratchx> in a browser.

2. Open the **Pac-Man** template for this project.

*Click Project templates -> Pac-Man*



3. Click the **full-screen** button, and then click the Green Flag



4. Play a few games of Pac-Man

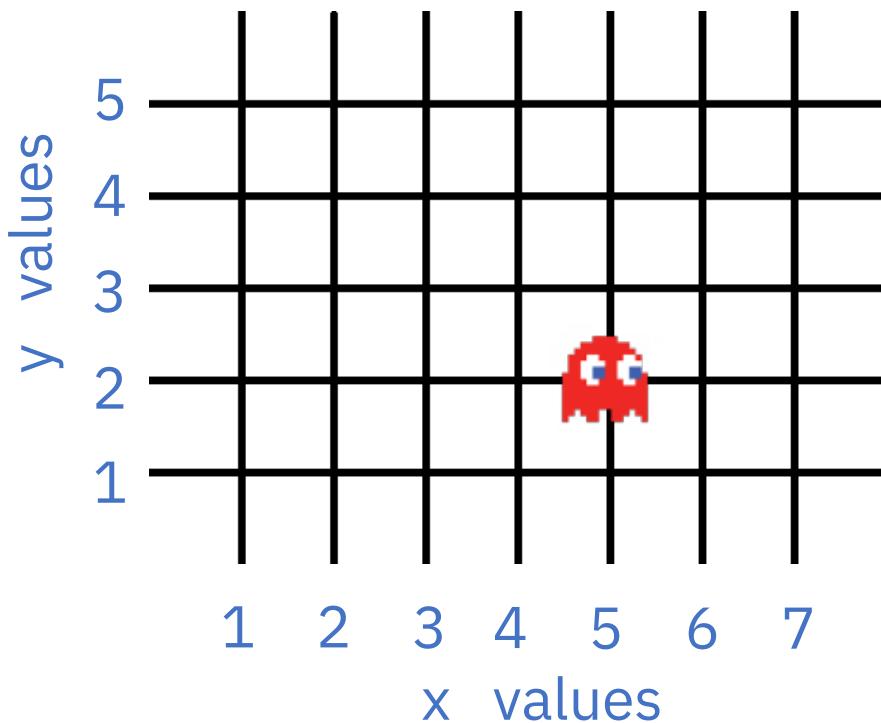
*You control Pac-Man, and have to avoid the ghost as long as you can.*

*Use the arrow keys to control Pac-Man's next move.*

*Click the green flag to start a new game.*

5. Try to come up with a plan for how Pac-Man can avoid the ghost

# Representing Pac-Man in Scratch



The game board is a graph, with both Pac-Man and the ghost only able to travel along lines.

The location of each character is stored as:

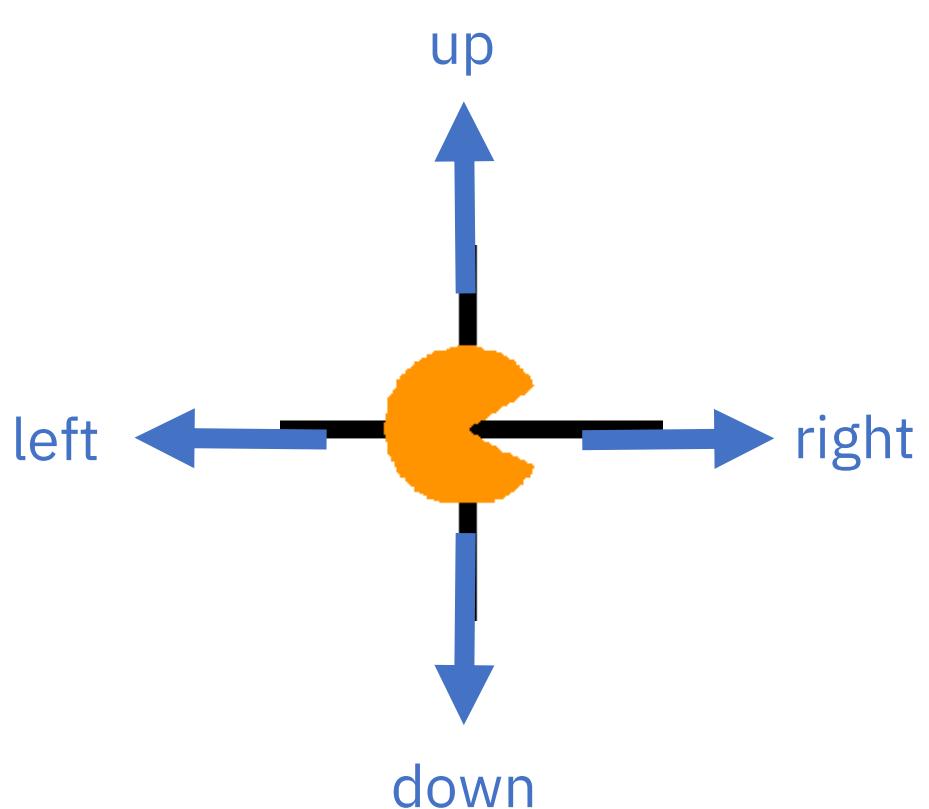
- \* an x-value (a number from 1 to 7)
- \* a y-value (a number from 1 to 5)

For example, the ghost shown on the left is at:

X = 5  
Y = 2

At each turn, each character has to choose between four possible moves: up, down, left, right.

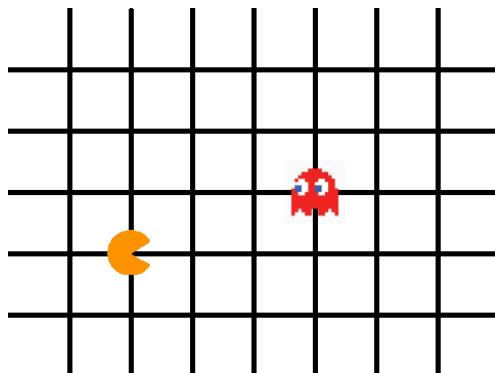
There are no diagonal moves.



## What are you going to do?

You're going to train Pac-Man to avoid the ghost. You'll do this by showing it examples of how you play the game.

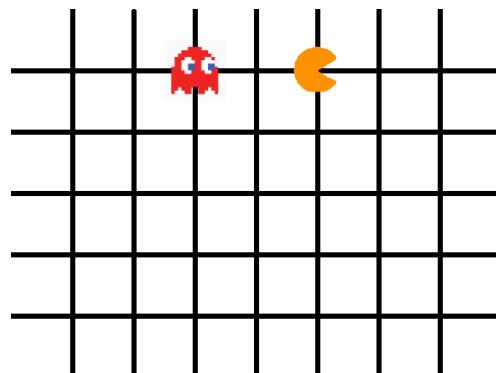
Imagine the board looks like this:



Imagine you decide to go up:

pacman x	2
pacman y	2
ghost x	5
ghost y	3
choice: up	

Imagine the board looks like this:



Imagine you decide to go down:

pacman x	5
pacman y	5
ghost x	2
ghost y	5
choice: down	

The computer will learn from the decisions that you make when you play the game.

That means if you make moves that avoid the ghost for a long time, the computer should learn how to avoid the ghost!

**6.** Close the Scratch window.

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

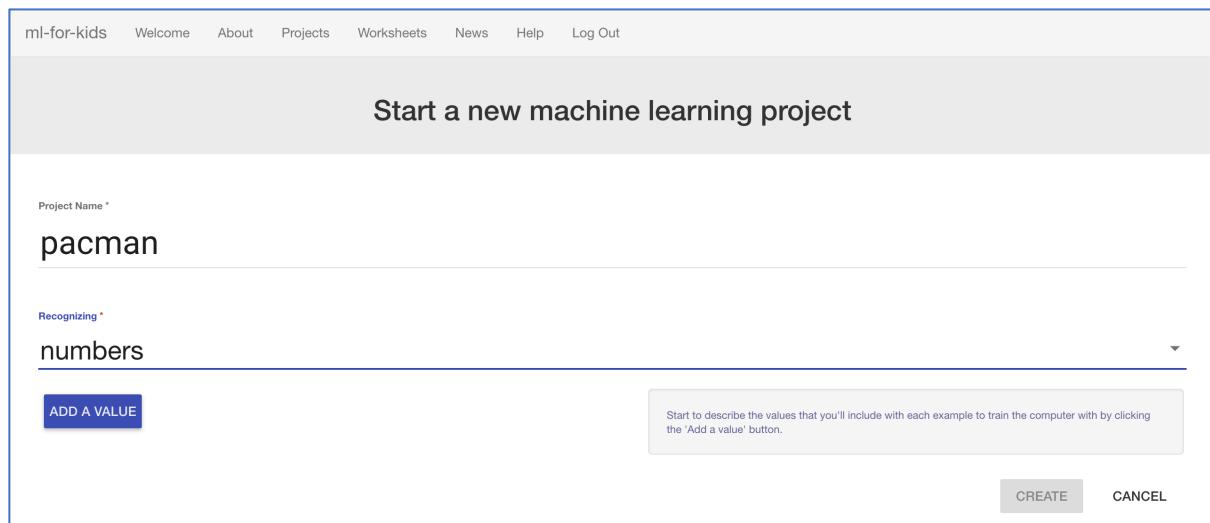
**8.** Click on “Get started”

- 9.** Click on “**Log In**” and type in your username and password  
*If you don't have a username, ask your teacher or group leader to create one for you.*  
*If you can't remember your username or password, ask your teacher or group leader to reset it for you.*

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

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

**12.** Name your project “pacman” and set it to learn how to recognise “numbers”



ml-for-kids   Welcome   About   Projects   Worksheets   News   Help   Log Out

Start a new machine learning project

Project Name \*

pacman

Recognizing \*

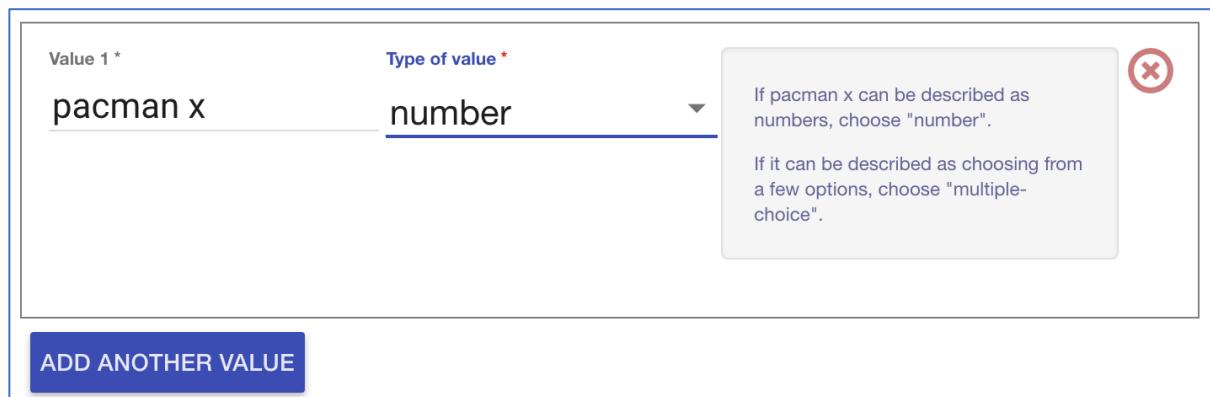
numbers

ADD A VALUE

Start to describe the values that you'll include with each example to train the computer with by clicking the 'Add a value' button.

CREATE   CANCEL

**13.** Click “**Add a value**” and name a value “pacman x” and make it a “number”.



Value 1 \*

pacman x

Type of value \*

number

If pacman x can be described as numbers, choose "number".  
If it can be described as choosing from a few options, choose "multiple-choice".

ADD ANOTHER VALUE

**14.** Click “Add another value” again and repeat to add values for the other three positions: “pacman y”, “ghost x”, “ghost y”

The screenshot shows the 'pacman' project configuration page. At the top, there's a 'Project Name' field containing 'pacman'. Below it, a 'Recognizing' dropdown menu is set to 'numbers'. Underneath, four input fields are arranged in a 2x2 grid:

Value 1 *	Type of value *
pacman x	number
Value 2 *	Type of value *
pacman y	number

Value 3 *	Type of value *
ghost x	number
Value 4 *	Type of value *
ghost y	number

Below these fields is a blue 'ADD ANOTHER VALUE' button. At the bottom right are 'CREATE' and 'CANCEL' buttons.

**15.** Click **Create**.

**16.** You should see “pacman” in the list of your projects. Click on it.

The screenshot shows the 'Your machine learning projects' page. At the top, there's a navigation bar with links: ml-for-kids, Welcome, About, Projects, Worksheets, News, Help, Log Out. Below the navigation, a title 'Your machine learning projects' is displayed. A button '+ Add a new project' is visible. Two projects are listed in cards:

- car or cup**  
Recognising **images** as **car or cup**
- pacman**  
Recognising **numbers**

Each card has a trash can icon for deletion.

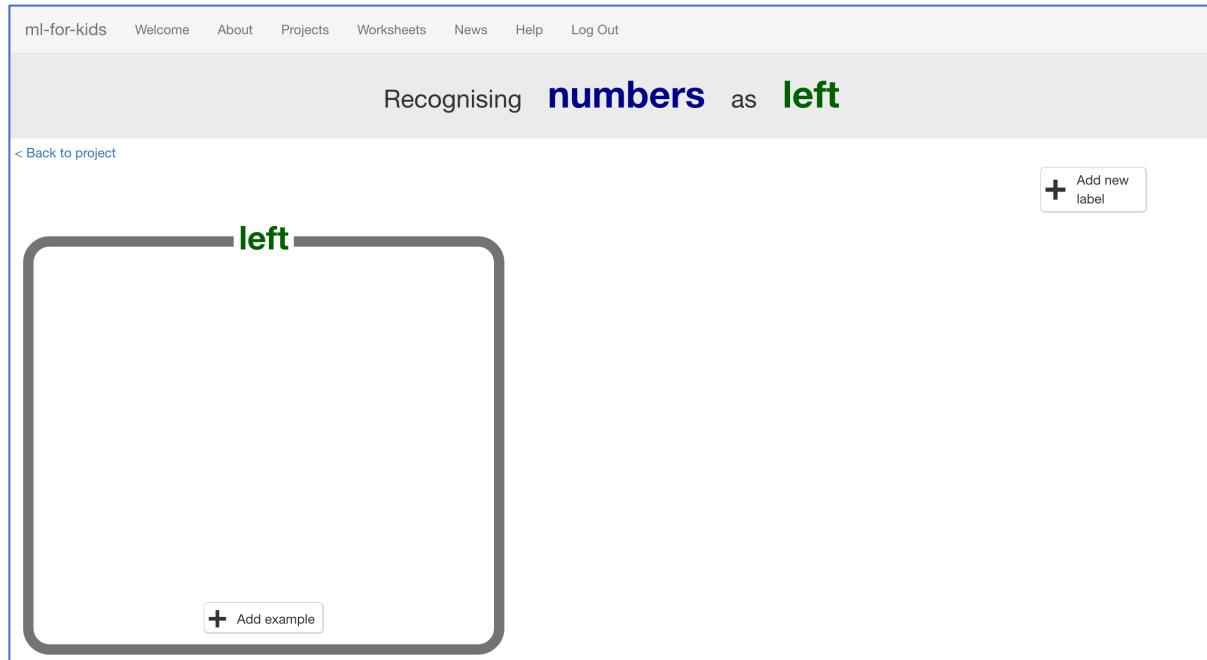
**17.** Click the “Train” button

The screenshot shows the 'Train' section of the project page. At the top, the project name 'pacman' is displayed. Below it are three buttons:

- Train**: Collect examples of what you want the computer to recognise. A 'Train' button is present.
- Learn & Test**: Use the examples to train the computer to recognise numbers. A 'Learn & Test' button is present.
- Scratch**: Use the machine learning model you've trained to make a game in Scratch. A 'Scratch' button is present.

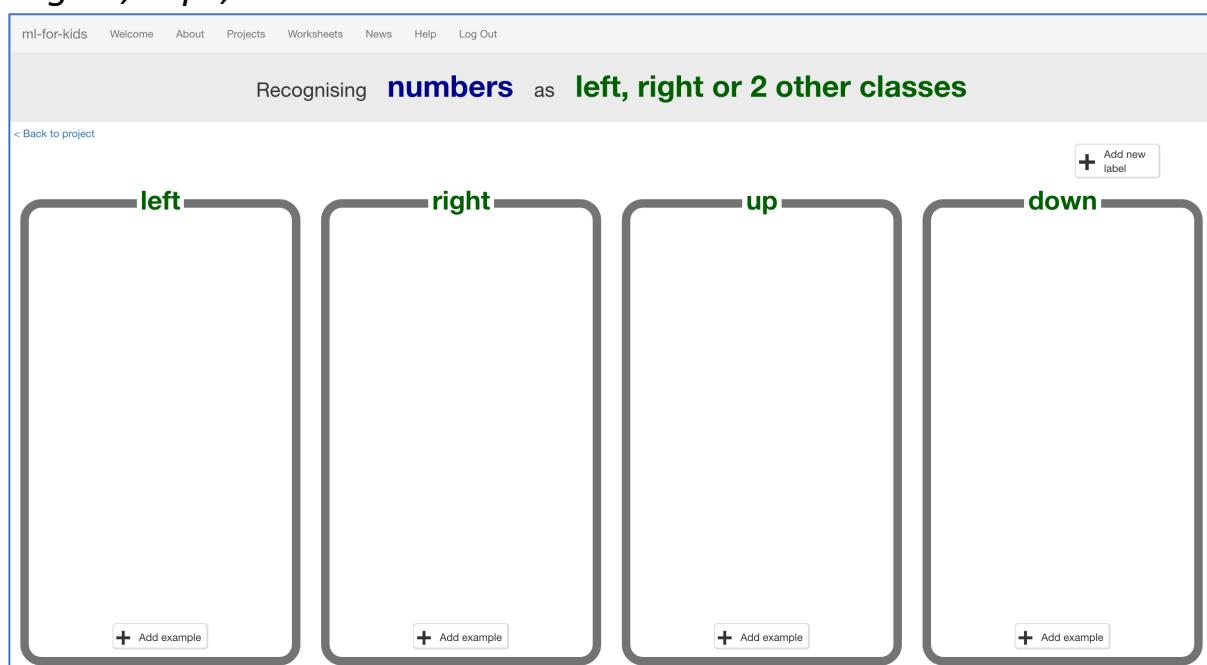
## 18. Click “+ Add new label” and create a label called “left”

*Examples of the locations of the Pac-Man and ghost when you go left will go in this bucket.*



## 19. Click “+ Add new label” again and create labels for the other three moves in the game.

“right”, “up”, “down”



## 20. Click the “< Back to project” link then click **Scratch**

## 21. Click the **Open in Scratch** button

*It will warn you that you haven't trained the computer yet – but that's okay, as you'll use Scratch to collect the training examples.*

*Click the “go straight into Scratch now” link.*

The screenshot shows the 'More Blocks' section of a project editor. On the left, there is explanatory text and a code example. On the right, there is a preview of the Scratch script editor with some blocks loaded. A legend at the bottom right explains the status indicator colors.

**Left Panel (Explanatory Text):**

- Your project will add these blocks to the **More Blocks** tab in Scripts.
- `recognise numbers pacman x [pacman y : ghost x : ghost y : (label)]`  
Put numbers in the input for this, and it will return the label that your machine learning model recognises it as.
- `recognise numbers pacman x [pacman y : ghost x : ghost y : (confidence)]`  
This will return how confident your machine learning model is that it recognises the type of numbers. (As a number from 0 - 100).
- `[left : right : up : down]`  
These blocks represent the labels you've created in your project, so you can use their names in your scripts.
- This means you can do something like this:

**Right Panel (Scratch Preview):**

It will look something like this - except with the name of your project.

The Scratch script editor shows the 'Scripts' tab selected. In the 'More Blocks' category, several blocks are visible, including ones for recognizing numbers and directions. A green dot next to the project name 'pacman' indicates the model is trained.

**Bottom Right (Status Legend):**

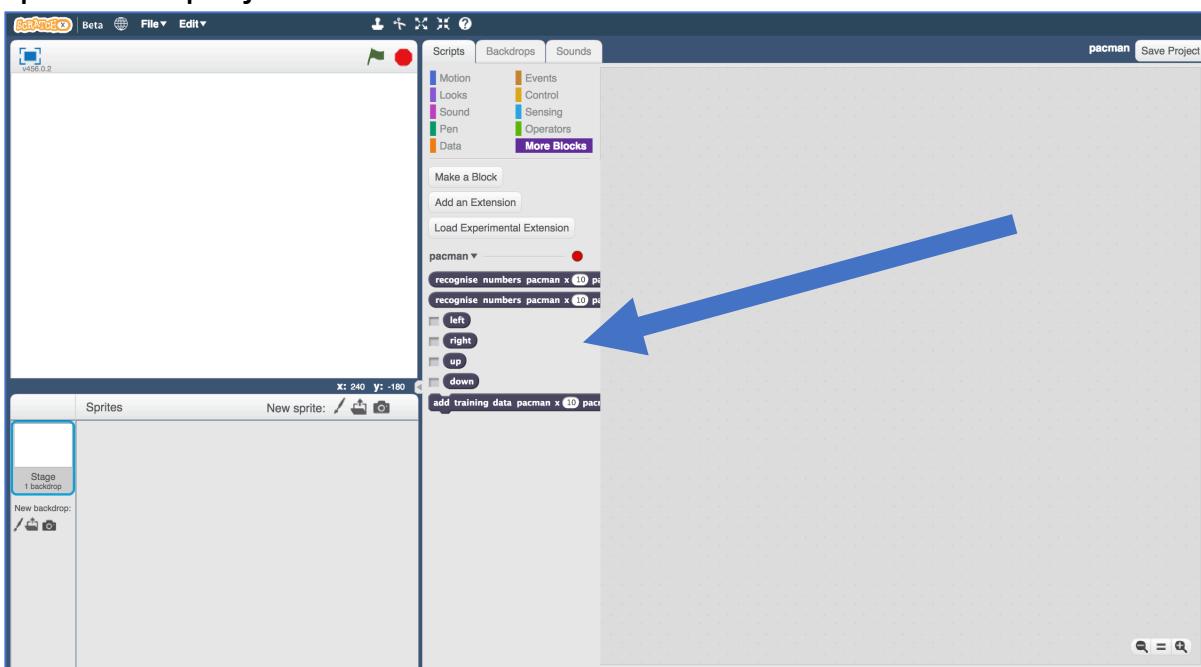
The coloured circle next to your project name tells you if your machine learning model is okay.

- ● means your model is trained and ready to go
- ● means your model hasn't finished training yet
- ● means something went wrong. Go back to the [Learn & Test](#) page to see what went wrong with training.

**Bottom Left (Buttons):**

[Open in Scratch](#)

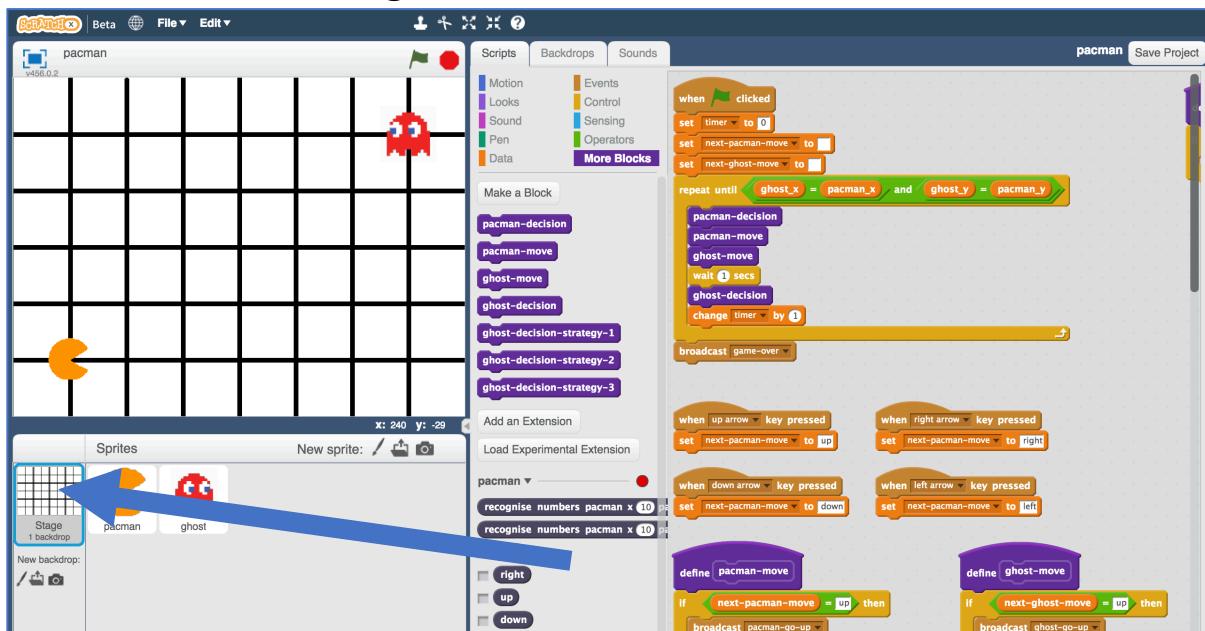
## 22. You should see new blocks in the “More blocks” section from your “pacman” project.



**23.** Open the Pac-Man template project again.

*Click Project templates -> Pac-Man*

**24.** Click on the “Stage”



**25.** Modify the key press scripts to use the move choices you created

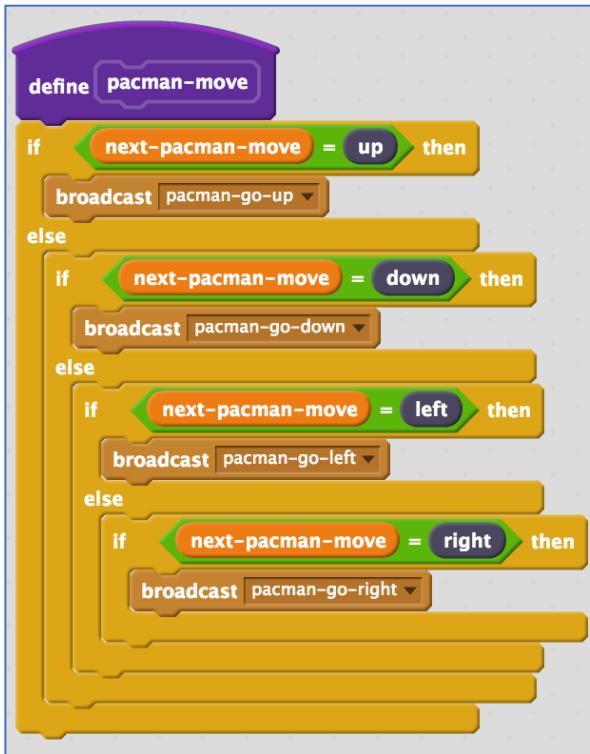
*Drag the “up”, “down”, “left” and “right” blocks from your “pacman” project into the boxes in the scripts like in the picture below.*

*You don’t need to add these scripts – you just need to drag in the dark blue “up”, “down”, “left”, “right” blocks from your project.*



**26.** Modify the custom block “pacman-move” to use the move choices you created

*Drag the “up”, “down”, “left” and “right” blocks from your “pacman” project into the boxes in the script like in the picture below.*



**27.** Update the custom block “pacman-decision” to add every move you make to your machine-learning training

*The purple “pacman-decision” block is ready for you in the starter project, at the top right of the Stage canvas.*



**28.** Train the computer!

*Click on full-screen again, and then the Green Flag.*

*Play a few games of Pac-Man, doing your best to avoid the ghost. The better you play, the better the computer has to learn from.*

## 29. Save your project

Click **File -> Save project**

Name the file something like “pacman-learn.sbx” to remind yourself that this version of the project is the one to train Pac-Man.

## What have you done so far?

You’re teaching a computer to play Pac-Man.

So far, you updated a Scratch Pac-Man game so that it can collect examples of how you play and add them to a set of examples. And you’ll use those examples to train a machine learning “model”.

## 30. Go back to the training tool

### 31. Click the “**< Back to project**” link, then click the “**Train**” button

*You should see the training examples you collected by playing Pac-Man.*

The screenshot shows a web interface for training a machine learning model. At the top, there's a navigation bar with links for ml-for-kids, Welcome, About, Projects, Worksheets, News, Help, and Log Out. Below the navigation is a title: "Recognising numbers as left, right or 2 other classes". A "Back to project" link is visible. On the right, there's a "Add new label" button with a plus sign. The main area contains four categories of training examples:

- left:** Contains 12 examples. Each example is a grid of three rows and three columns. The first row contains "pacman x: 6", "pacman y: 4", "ghost x: 6", and "ghost y: 2". The second row contains "pacman x: 4", "pacman y: 3", "ghost x: 5", and "ghost y: 3". The third row contains "pacman x: 5", "pacman y: 2", "ghost x: 5", and "ghost y: 1". Buttons at the bottom say "+ Add example".
- right:** Contains 12 examples. Each example is a grid of three rows and three columns. The first row contains "pacman x: 1", "pacman y: 2", "ghost x: 6", and "ghost y: 5". The second row contains "pacman x: 3", "pacman y: 1", "ghost x: 4", and "ghost y: 3". The third row contains "pacman x: 5", "pacman y: 2", "ghost x: 2", and "ghost y: 3". Buttons at the bottom say "+ Add example".
- up:** Contains 12 examples. Each example is a grid of three rows and three columns. The first row contains "pacman x: 2", "pacman y: 1", "ghost x: 7", and "ghost y: 4". The second row contains "pacman x: 6", "pacman y: 1", "ghost x: 4", and "ghost y: 1". The third row contains "pacman x: 5", "pacman y: 1", "ghost x: 2", and "ghost y: 1". Buttons at the bottom say "+ Add example".
- down:** Contains 12 examples. Each example is a grid of three rows and three columns. The first row contains "pacman x: 3", "pacman y: 2", "ghost x: 5", and "ghost y: 4". The second row contains "pacman x: 2", "pacman y: 3", "ghost x: 3", and "ghost y: 4". The third row contains "pacman x: 6", "pacman y: 4", "ghost x: 5", and "ghost y: 5". Buttons at the bottom say "+ Add example".

## 32. Click the “**< Back to project**” link, then click “**Learn & Test**”

### 33. Click the “Train new machine learning model” button

Machine learning models

< Back to project

**What have you done?**

You've collected examples of numbers for a computer to use to recognise when numbers are left, right or 2 other classes.

You've collected:

- 23 examples of down,
- 17 examples of left,
- 34 examples of right,
- 33 examples of up

**What's next?**

Ready to start the computer's training?

Click the button below to start training a machine learning model using the examples you've collected so far.

(Or go back to the Train page if you want to collect some more examples first.)

Info from training computer:

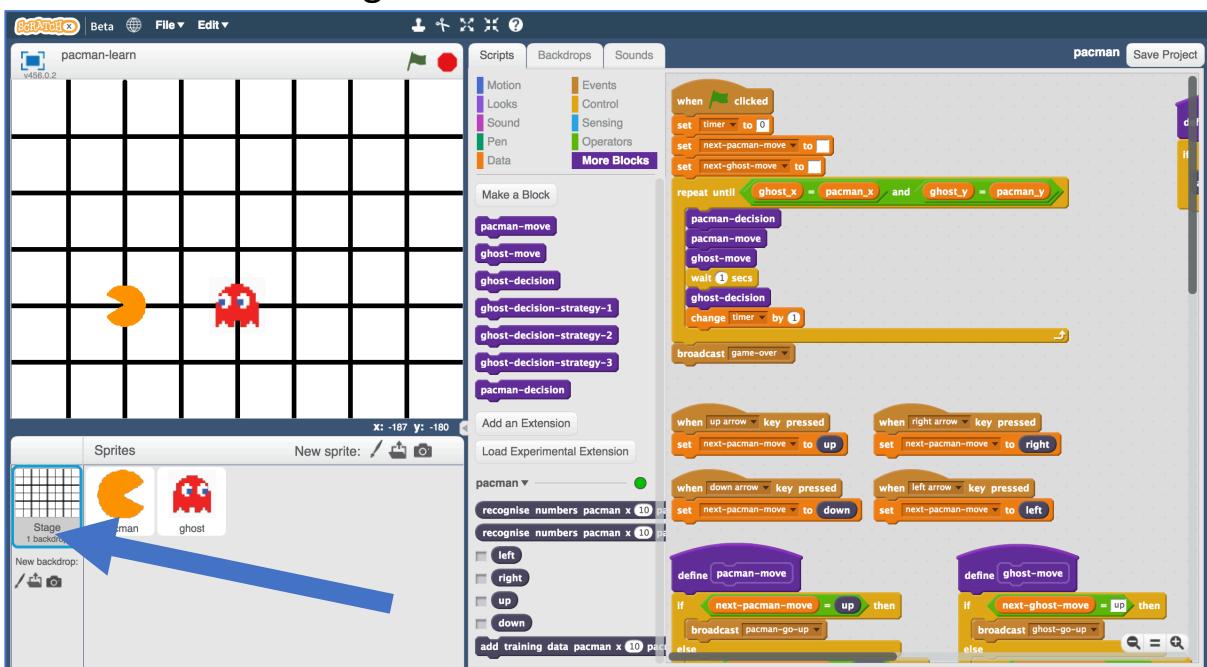
Train new machine learning model

### 34. Switch back to the Scratch window.

*If you accidentally closed it, you can get back to it by doing this:*

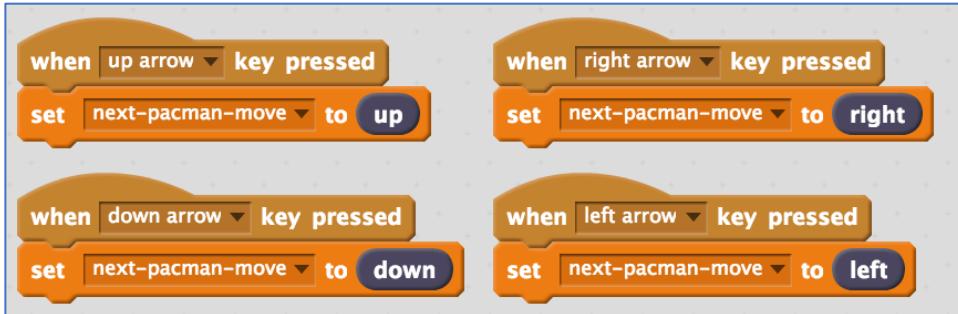
- \* Click the “< Back to project” link
- \* Click the “Scratch” button
- \* Click the “Open in Scratch” button
- \* Open the Scratch project you saved before, with “File” -> “Load Project”

### 35. Click on the Stage



**36.** Delete the key press scripts because it's the computer's turn!  
*(Delete a script by dragging it back onto the palette, or right-click and choose "Delete")*

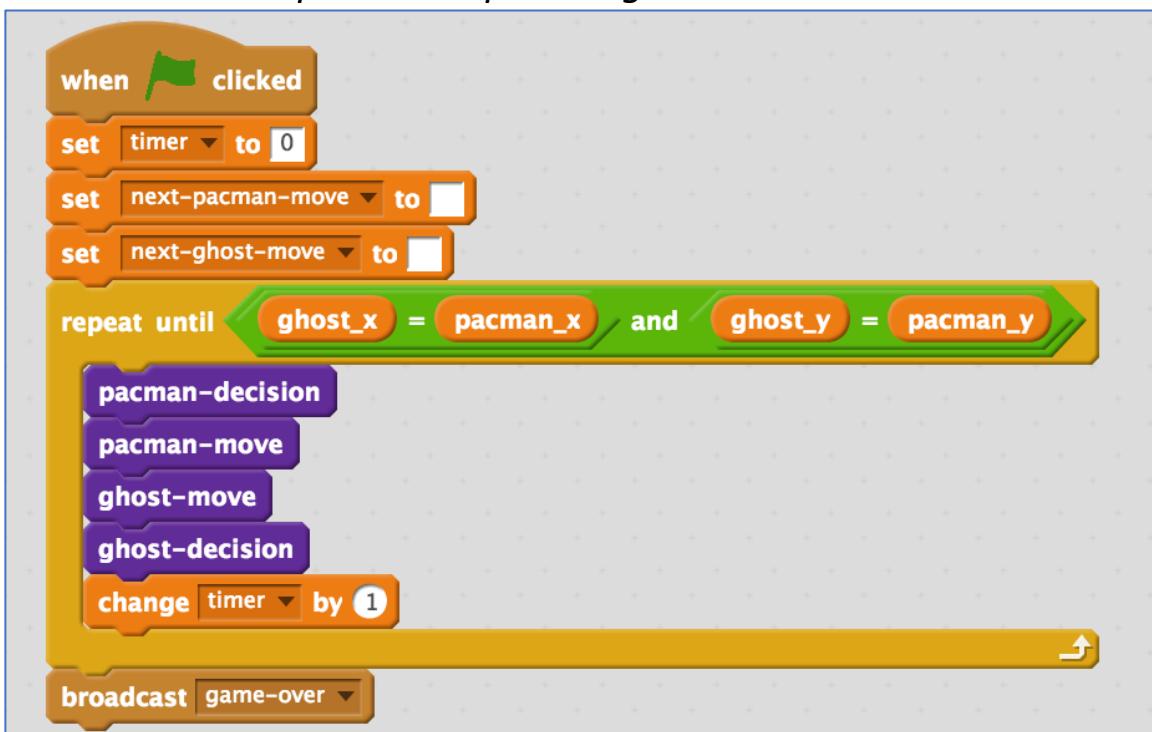
*These are the scripts you don't need any more:*



**37.** Modify the custom "pacman-decision" block  
*Instead of learning from what you are doing, now you want to use your machine learning model*



**38.** Modify the "Click Green Flag" script to remove "wait 1 second".  
*You want the script to end up looking like this:*



**39.** Save your project

*Click **File** -> **Save project***

*Name the file something like “pacman-play.sbx” to remind yourself that this version of the project is where the computer controls Pac-Man.*

**40.** Test the computer!

*Click on full-screen again, and then the Green Flag.*

*Watch the Pac-Man you’ve trained try to avoid the ghost.*

**41.** Open the training project “pacman-learn.sbx”.

*Make sure you save your pacman-play project first!*

*Click **File** -> **Load Project***

**42.** Train the computer some more by playing a few more games.

**43.** Go back to the training tool

**44.** Go back to the “Learn & Test” page

*Click the “< Back to project” link, and then click “Learn & Test”*

**45.** Click the “Train new machine learning model” button again

**46.** Switch back to the Scratch window.

*If you accidentally closed it, you can get back to it by doing this:*

- \* *Click the “< Back to project” link*
- \* *Click the “Scratch” button*
- \* *Click the “Open in Scratch” button*

**47.** Open the testing project “pacman-play.sbx”

*Click **File** -> **Load Project***

**48.** Test the computer again

*Did the computer do any better after more training?*

## Tips

### **Getting stuck in a loop**

Sometimes the computer can get lucky, and find a circular route around the board that gets into a never-ending loop.

When this happens, Pac-Man will never lose!

You can press the red stop button if you need to stop though.

### **Keep training**

The more examples the computer has to learn from, the better it will get. If you have time, play a lot of games and train a new model again.

### **Don't be kind!**

You might be tempted to go easy on the ghost when you're playing against it.

Don't. It is learning from the way that you play. If you don't complete a three-in-a-row when you can, you will be teaching it that it should do that.

If you want it to get better quickly, **play as well as you can**.

## What have you done?

You've trained a computer to play Pac-Man.

You didn't have to describe the rules to the computer.

You didn't tell it that it should try to avoid the ghost.

You didn't describe the boundaries of the board.

(The rules are in the Scratch game, but that doesn't count – that wasn't used in the machine learning model).

Instead, you showed it how you play, by collecting examples of decisions that you made when you play.

## Ideas and Extensions

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

Or come up with one of your own?

### Add another ghost

The game is beatable with only one ghost – Pac-Man can just carry on avoiding the ghost forever.

But with a second ghost chasing after Pac-Man, it will get really hard.

### Change the game board

Try making the game board bigger.

Or add obstacles that Pac-Man and the ghost will need to go around.