technocamps

technocamps

Machine Learning

Activity: What is Machine Learning (ML)?

Machine Learning

Machine Learning is a system with the ability to automatically learn and improve from experience without being explicitly programmed.

Machine Learning focuses on the development of computer programs that are provided with data and use it to learn by themselves.

Activity: Do You Know Any Machine Learning Systems?

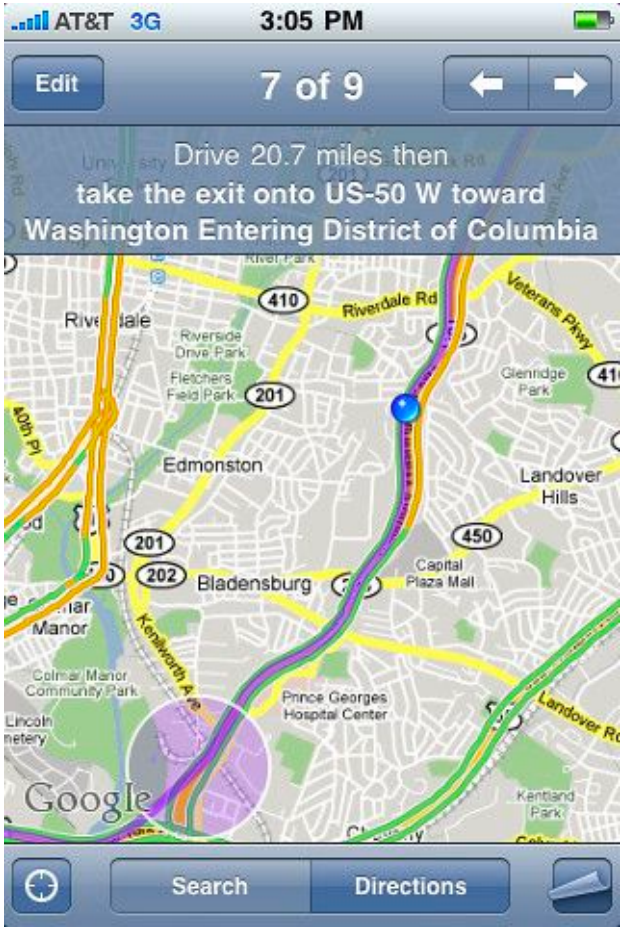
Current Machine Learning Systems
Siri

- Siri is a voice-activated assistant.
- Siri interprets your voice instructions, and, when possible, carries them out.
- Siri can open apps, give you movie times and sports scores, call or send messages to people in your contact list

Current Machine Learning Systems
Google Maps

Google can use user's location data for things like:

- real-time traffic updates,
- estimating current traffic speeds, and
- adjust directions accordingly.



Current Machine Learning Systems Security Cameras

- It is not possible for humans to keep monitoring hundreds/thousands of monitors at the same time, using technologies like:
- **object recognition** and **facial recognition**, this becomes possible.

Current Machine Learning Systems Dota 2

- Dota 2 is a strategy based video game.
- **OpenAI 5**, developed by the company OpenAI, backed by Elon Musk, has beaten pro-level Dota 2 players in one-on-one matches.
- Also beating amateur Dota 2 teams.

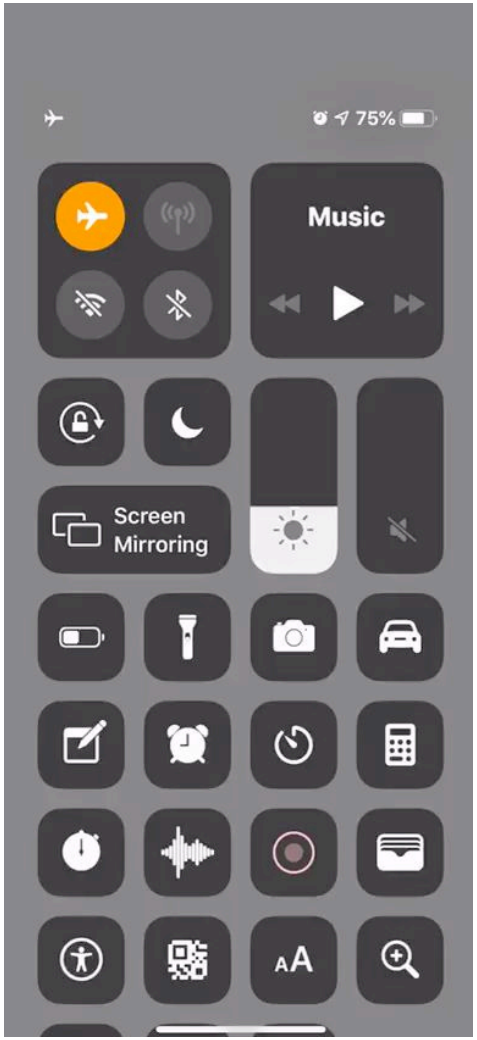
Current Machine Learning Apps / Games

Draw It - mobile app.

Players competes with each other in a online real time drawing game.

The goal is to draw one of the two categories given, so that the computer can make an accurate guess.

The computer is taught through ML to recognise drawings.

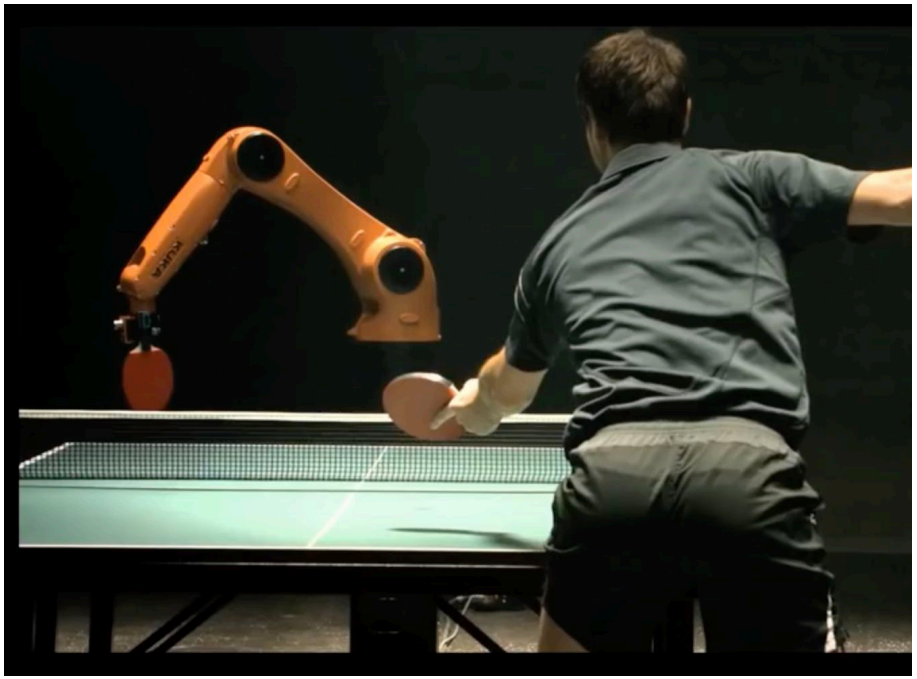


Click to play video.

Current Machine Learning Systems
KUKA

KUKA is one of the world's leading suppliers of robotics.

Table tennis bot



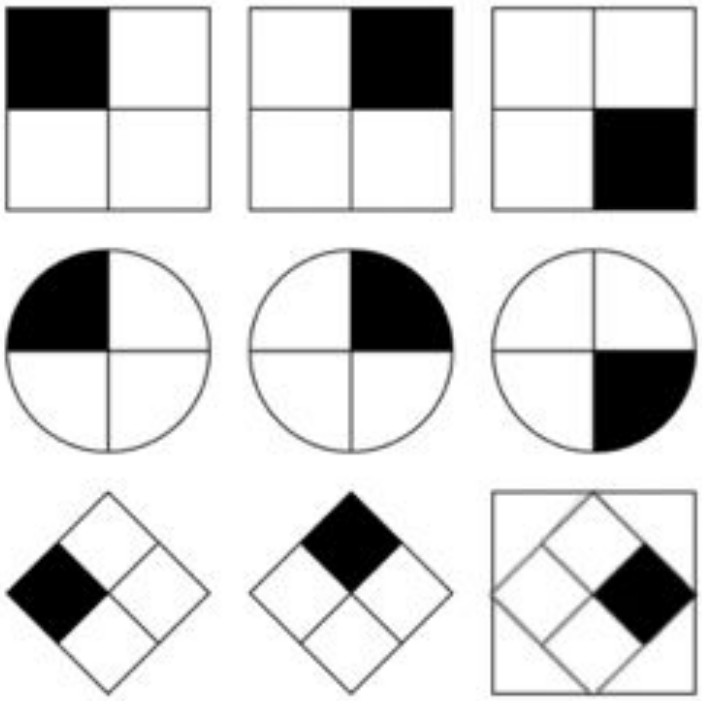
KUKA

Click to play video.

Activity: How Does Machine Learning Work?

Pattern Recognition

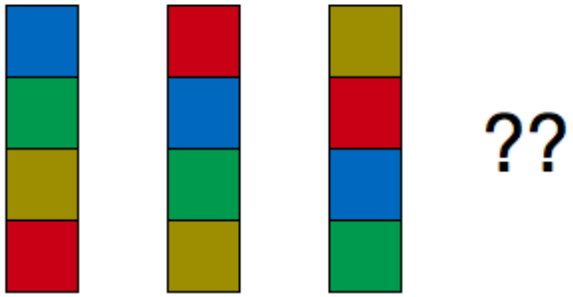
Pattern recognition is the ability to recognise patterns in data sets.

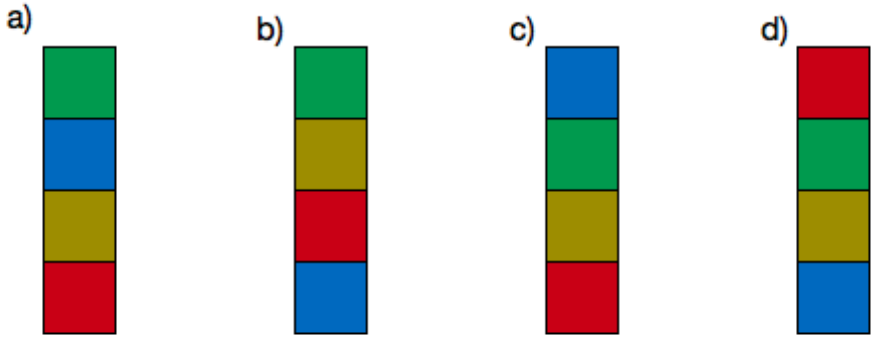


Activity: Pattern Recognition

Activity: Pattern Recognition (1)

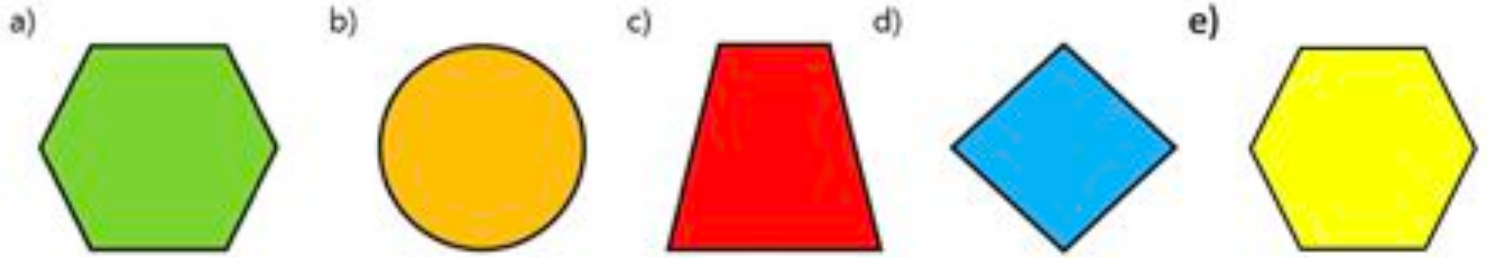
What comes next?





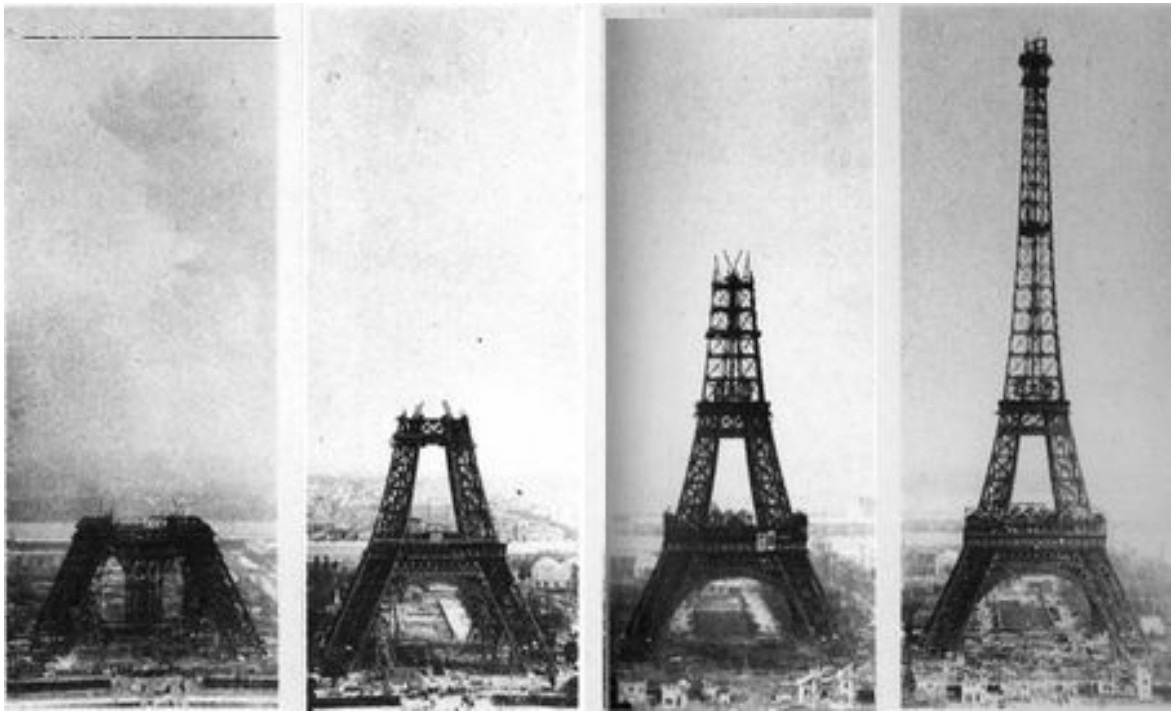
Activity: Pattern Recognition (2)

What comes next?



Activity: Pattern Recognition (3)

The Eiffel Tower was the tallest structure in the world until 1930. If the first three images above were taken on March 1888, July 1888, and November 1888, which of the following dates is the most reasonable for the final picture of the completed tower?



a) January 1890

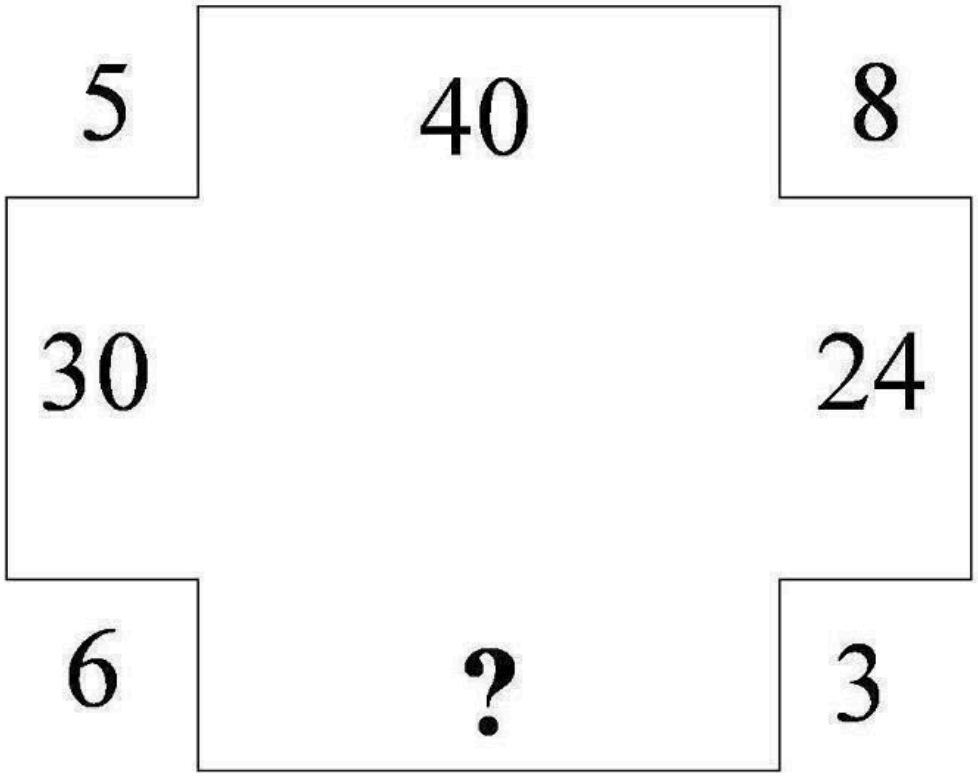
b) July 1889

c) March 1889

d) December 1888

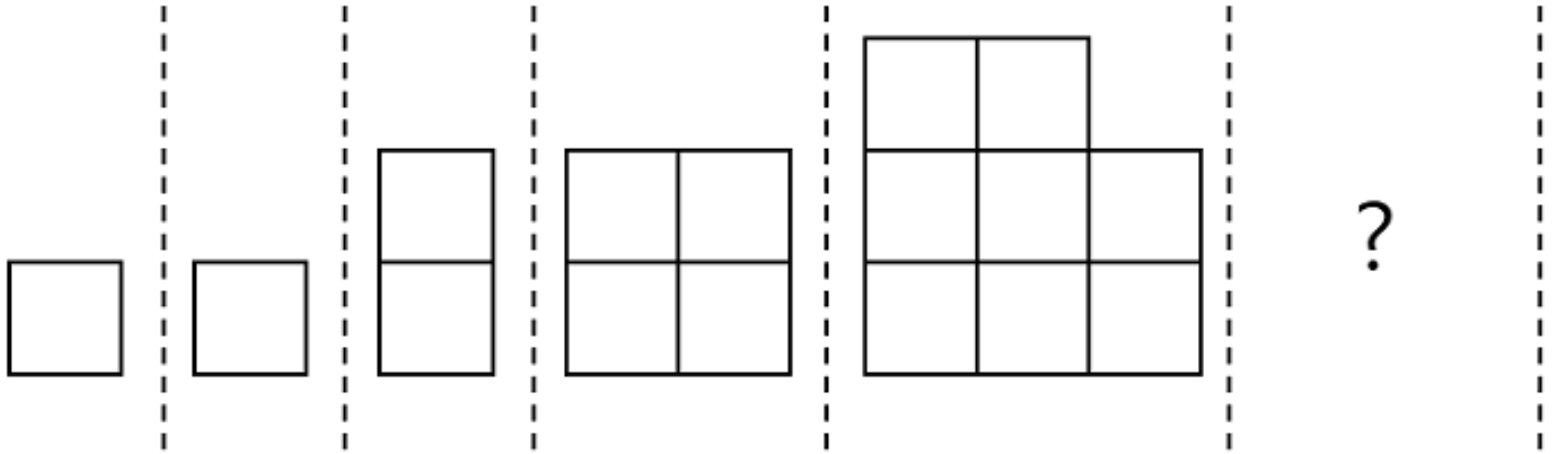
Activity: Pattern Recognition (4)

What number fits into the question mark?



Activity: Pattern Recognition (5)

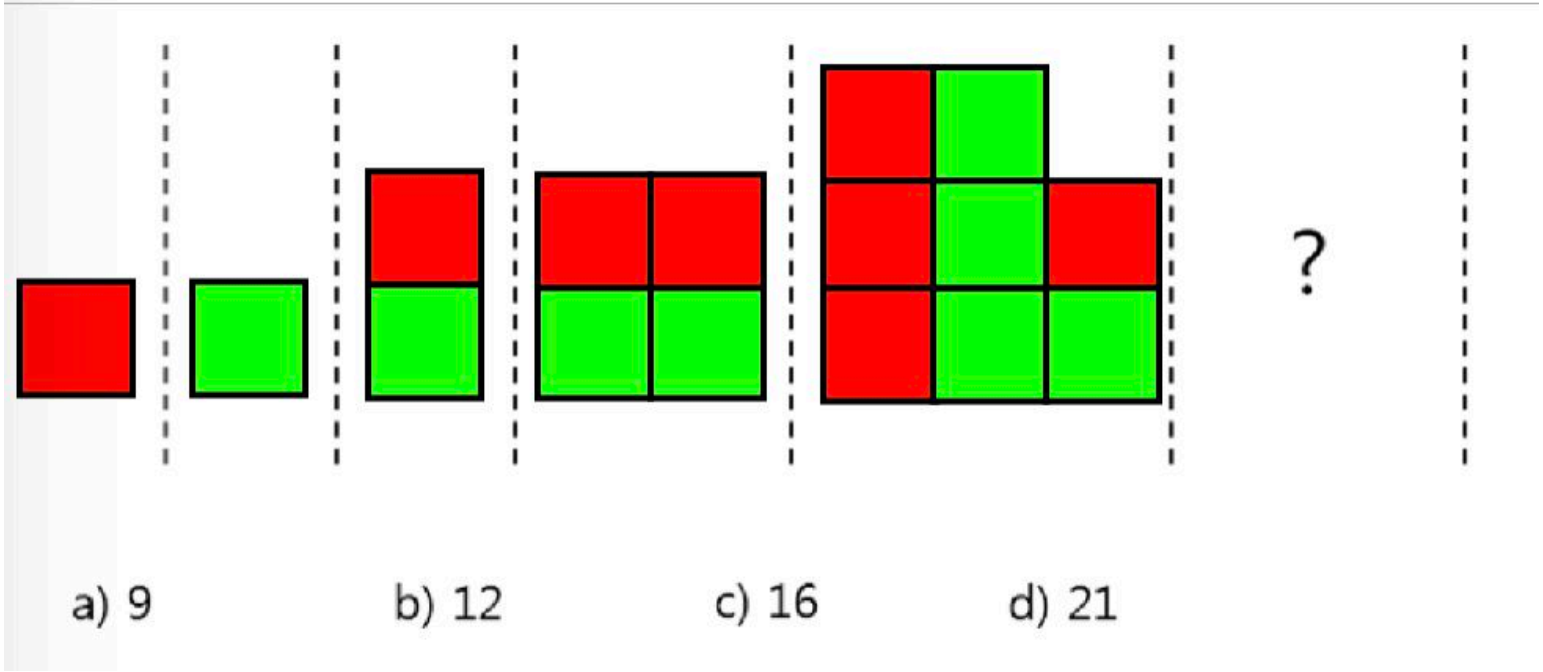
What is the number of squares in the last section where the question mark is?



a) 9 b) 12 c) 16 d) 21

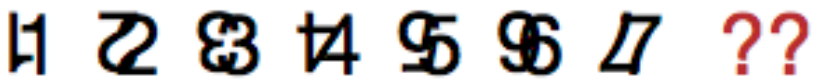
Activity: Pattern Recognition (5) Hint

What is the number of squares in the last section where the question mark is?



Activity: Pattern Recognition (6) Tricky

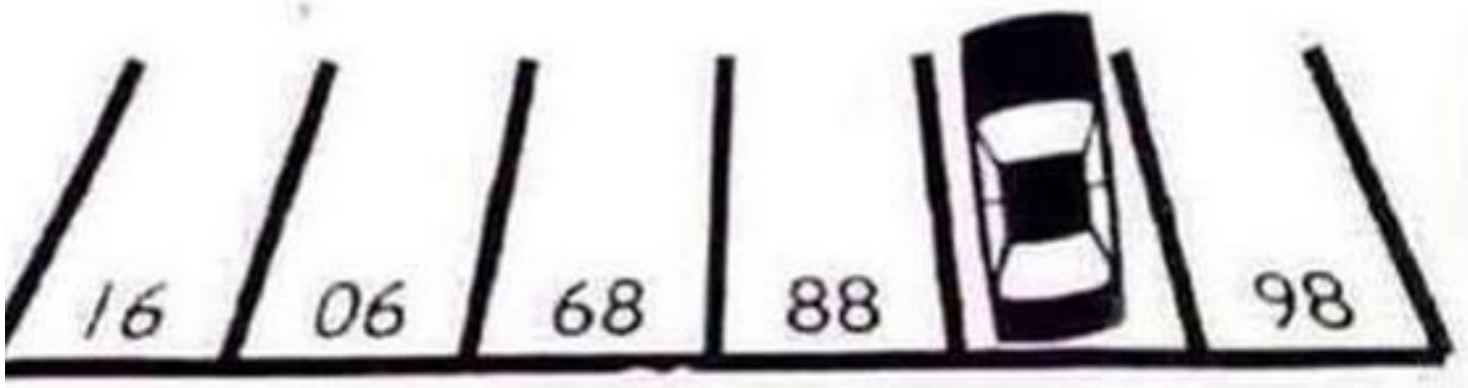
What comes next?



a) b) c) d)

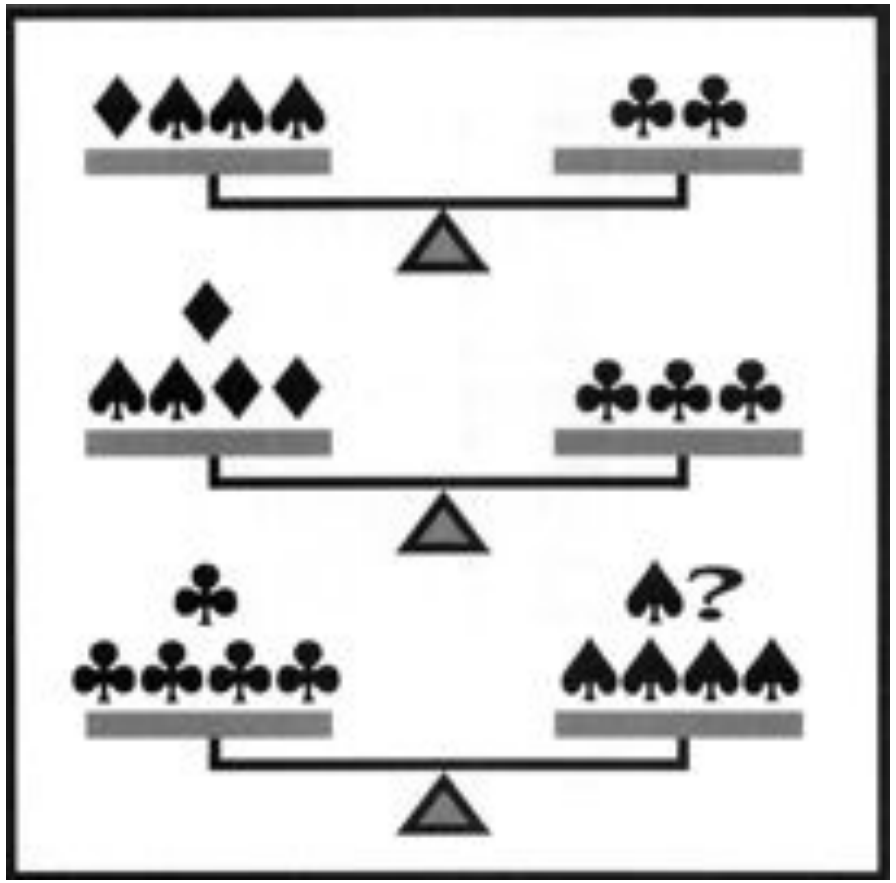
Activity: Pattern Recognition (7) Tricky

What number is this car parked in?



Activity: Pattern Recognition (8) Advanced

The top two scales in the image at the right are in perfect balance. How many diamonds will be needed to balance the bottom set?



Activity: ML Simulation

technocamps

Machine Learning in Scratch

https://machinelearningforkids.co.uk

Machine Learning Process

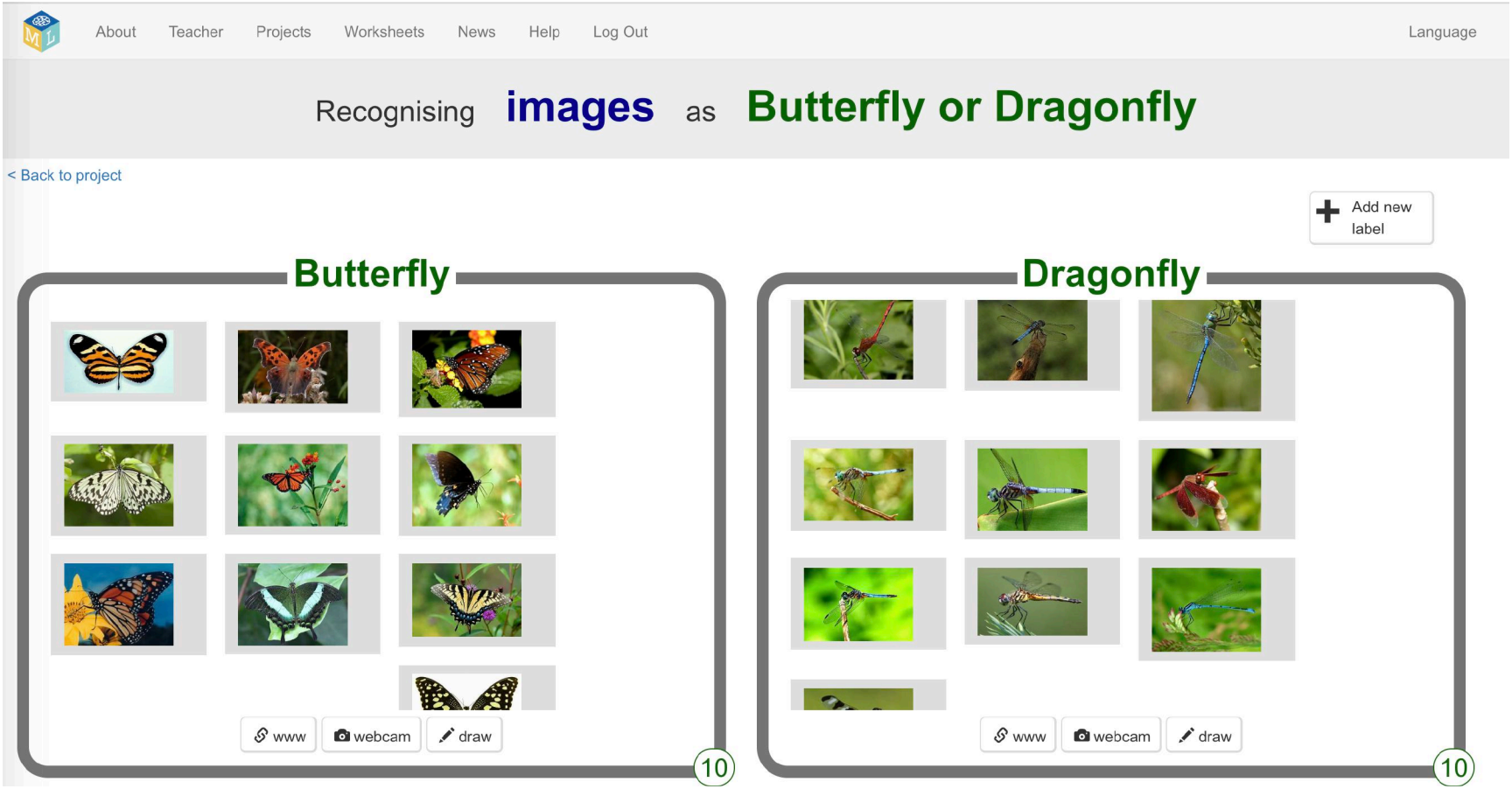
Let us say, we want to be able to train a computer to sort a set of photos into two piles: one pile of photos of butterflies, and one pile of photos of dragonflies.

How would we start this process?

What do we need?

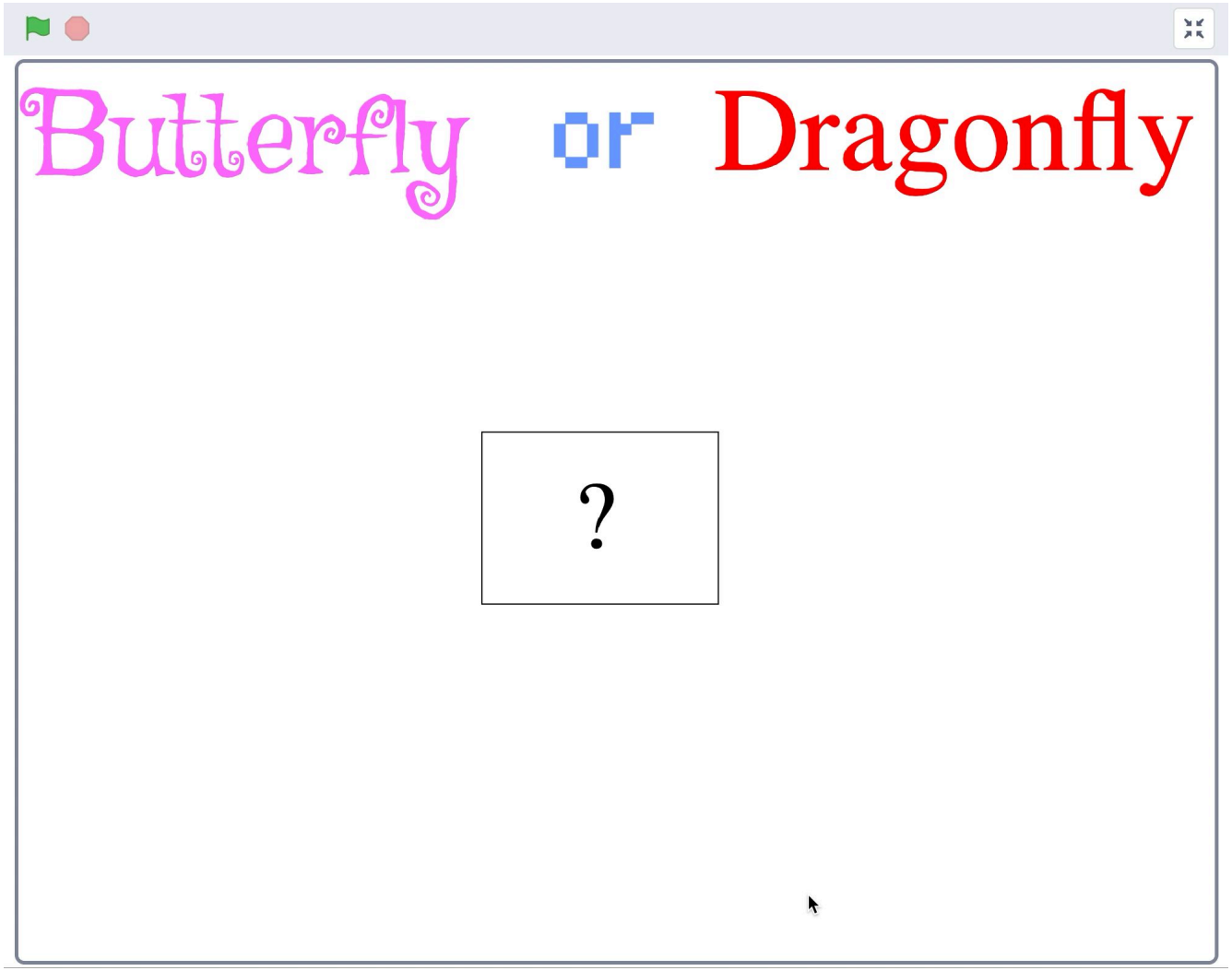
technocamps

The Training Data



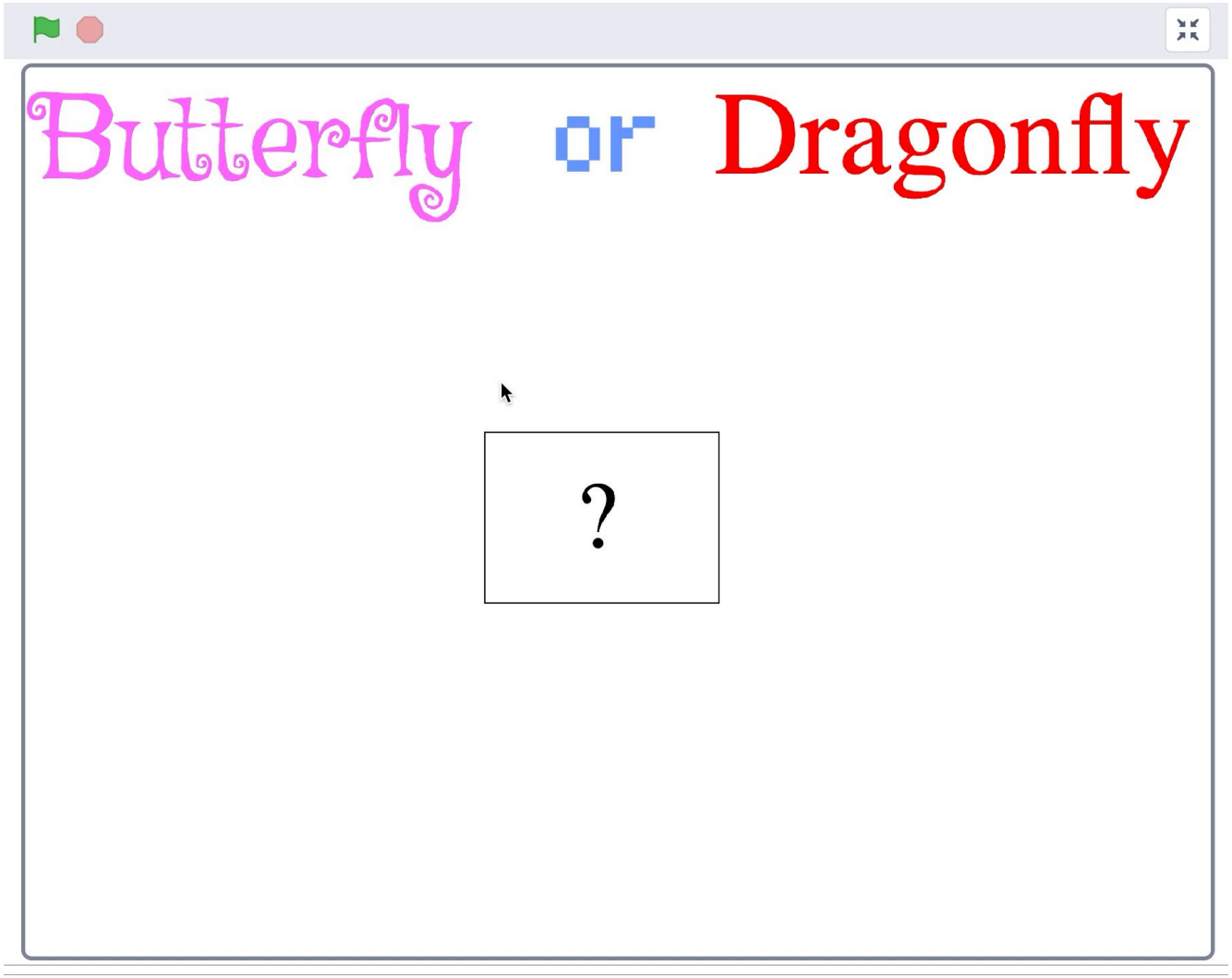
Without ML

Click to play video.

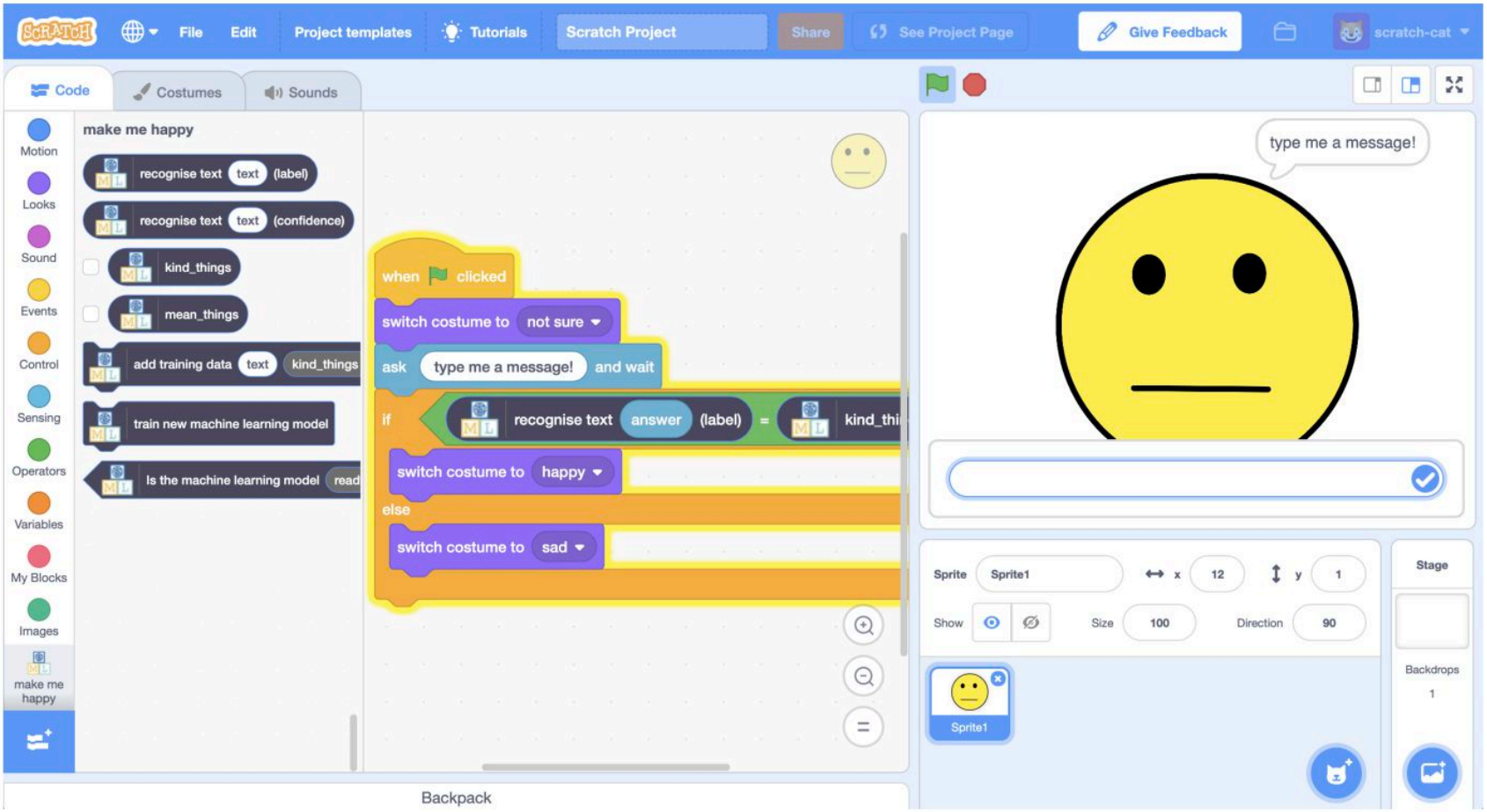


With ML

Click to play video.



Make Me Happy



Make Me Happy

Go to https://machinelearningforkids.co.uk/

Click on “Get started”.

Click on “Try it now”.

Click on “Projects” on the top menu bar.

Click the “+Add a new project” button.

Name your project “make me happy” and set it to learn how to recognise “text” and click the “Create” button.

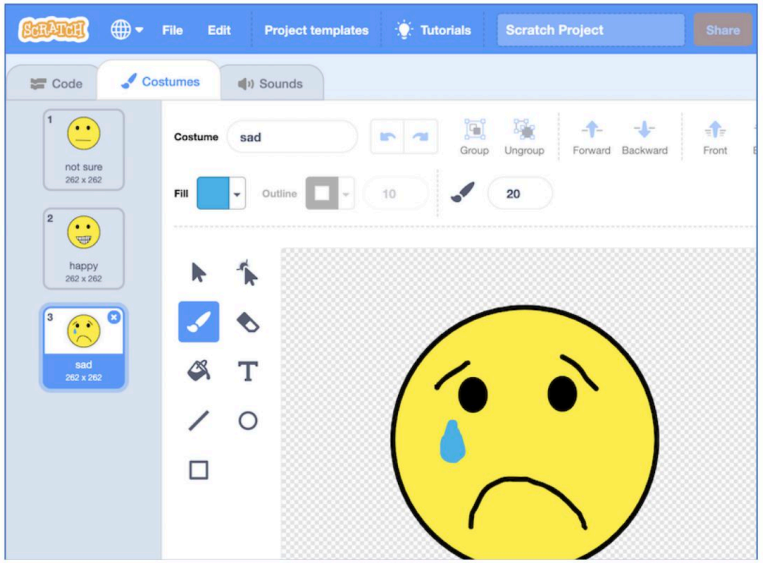
Make Me Happy

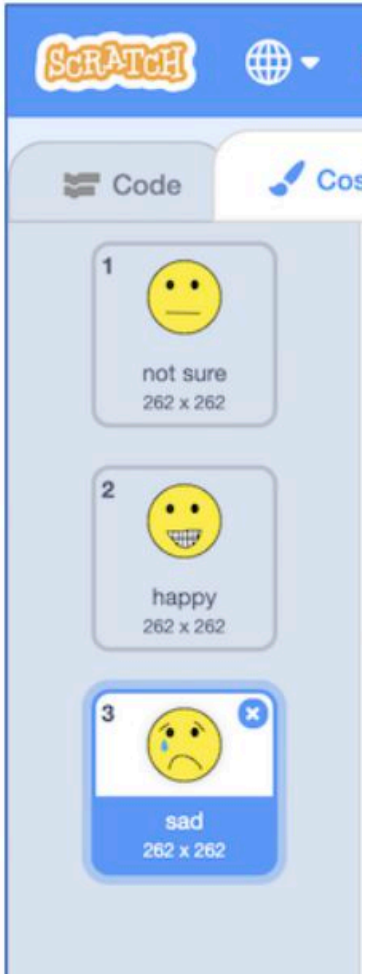
Make a sprite with three costumes:

1) "not sure" face.

2) "happy" face.

3) "sad" face.



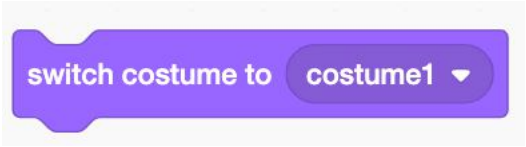


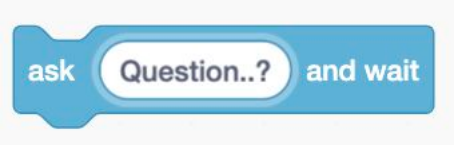
Make Me Happy

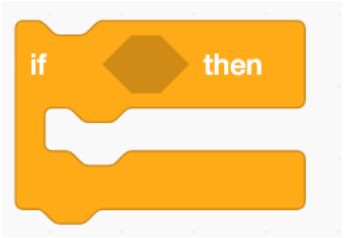
Program your sprite so that:

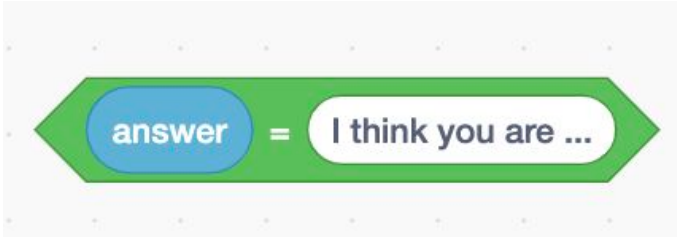
1) When we start the game, display not sure face.
2) Ask the question “Type a message: ”
3) If the answer is “I think you are cool” then, display happy face.
3) If the answer is “You smell bad” then, display sad face.

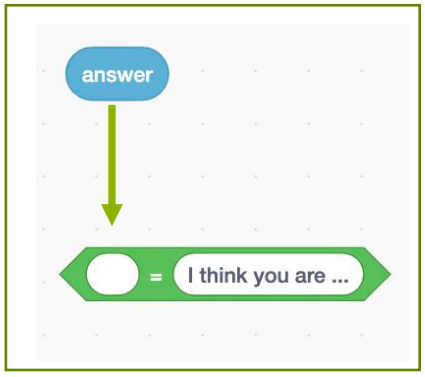
Make Me Happy



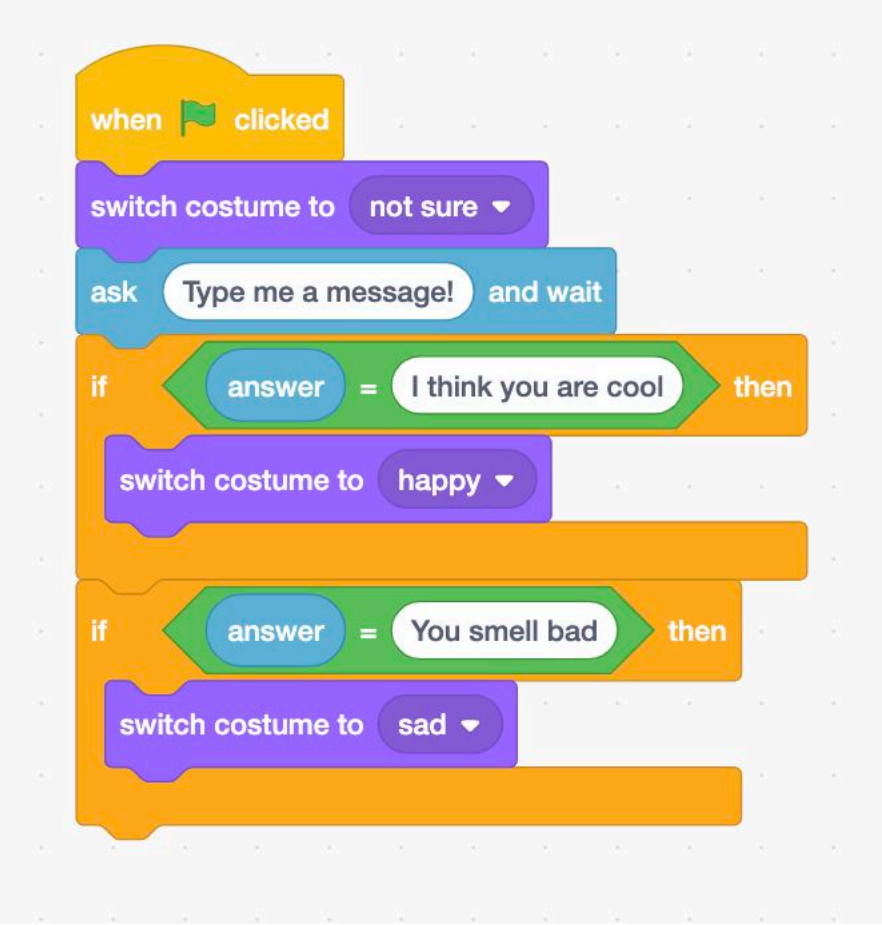








Make Me Happy



What you have done so far?

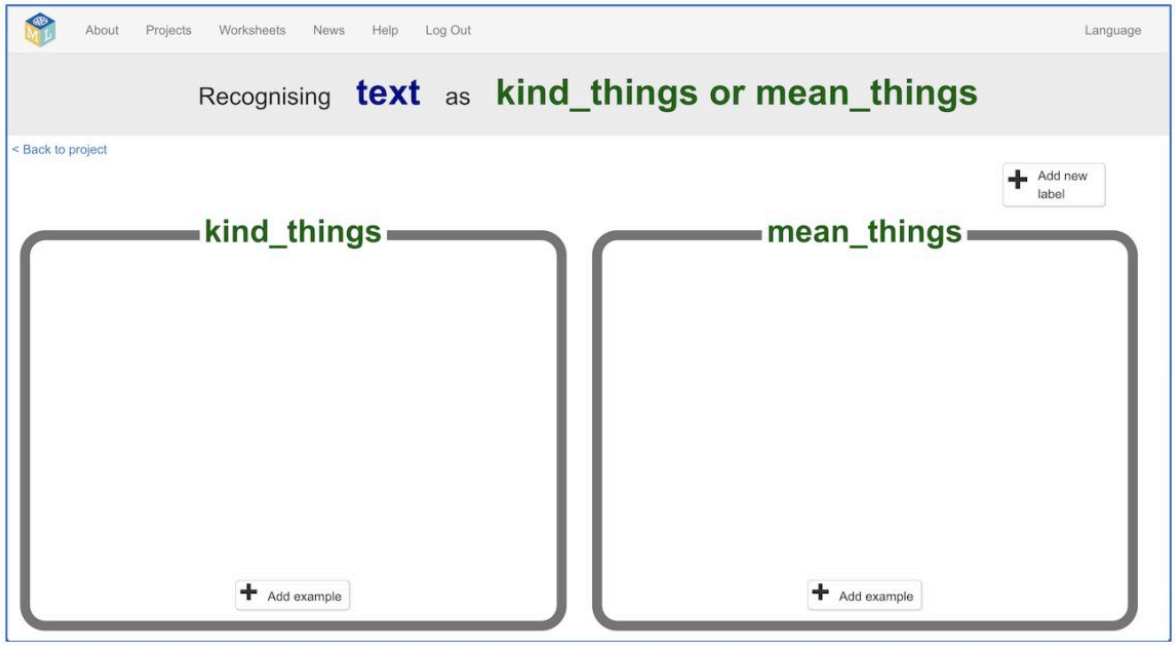
You have created a sprite that is happy when you say that it is ‘cool’ and sad when you say it ‘smells bad’.

However, what is the problem with our game?

How can we improve it?

Make Me Happy

Training the machine



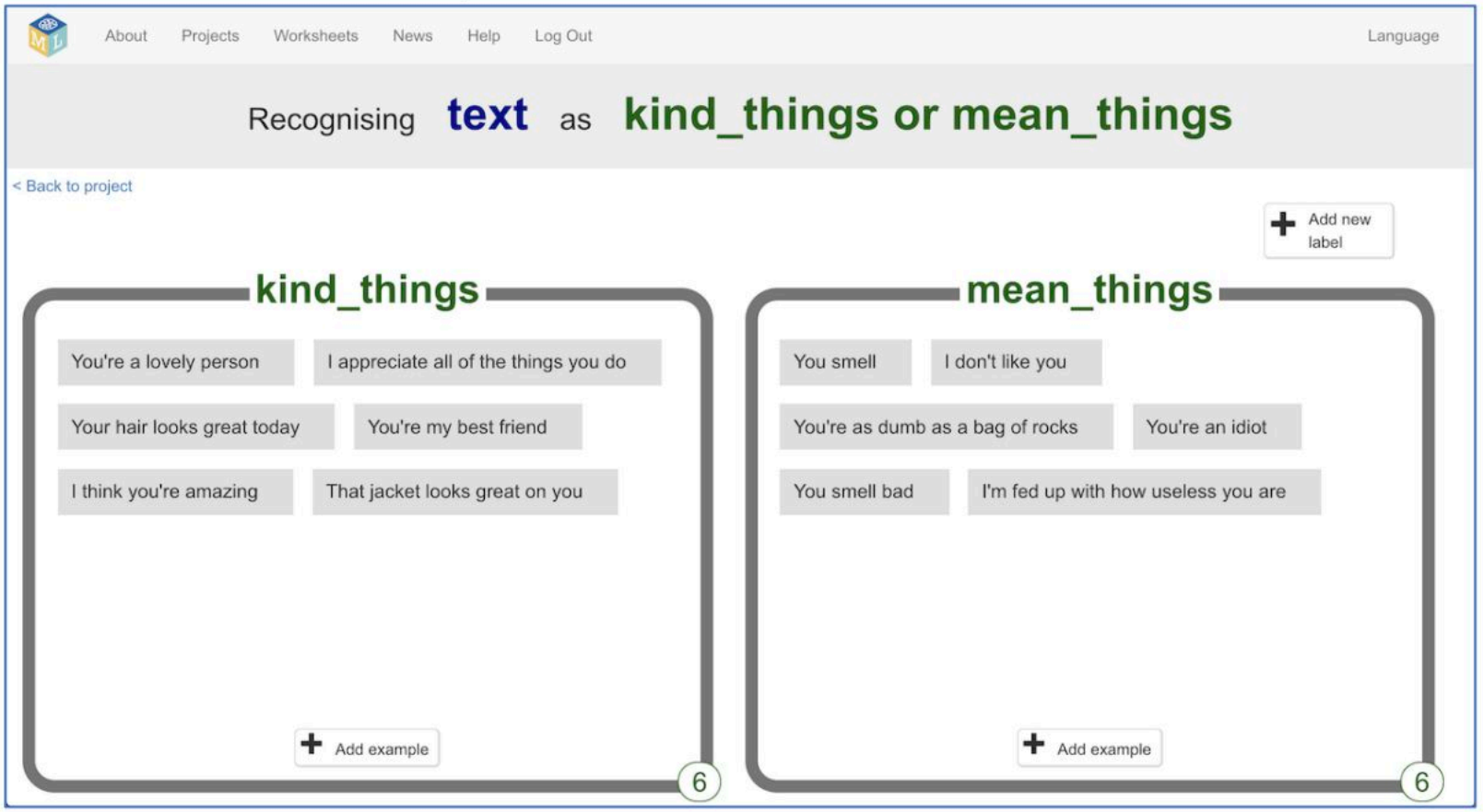
Make Me Happy

Click the "Add example" button in the "kind things" label, and type in the nicest, kindest compliment you can think of.

Click the "Add example" button in the "mean things" label, and type in the meanest, cruelest insult you can think of *(**be reasonable and appropriate**)*.

Write at least **six** compliments and at least **six** insults.

Make Me Happy



Make Me Happy

Click on the "< Back to project" link.

Then click on the "Learn & Test" button.

Click on the "Train new machine learning model" button. *As long as you have entered an **even** amount of examples, the computer should start to learn how to recognise messages from the examples you have given it.*

What Have You Done So Far?

You have started to train a computer to recognise text as being kind or mean.

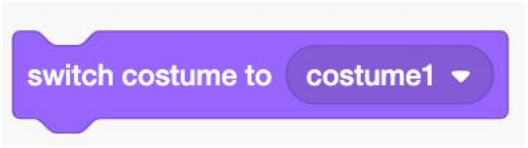
This is called "supervised learning" because of the way you are supervising the computer's training. The computer will learn from patterns in the examples you have given it.

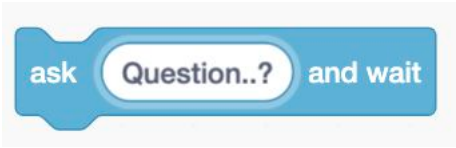
Make Me Happy

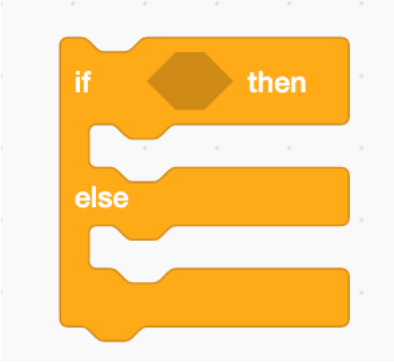
Program your sprite so that:

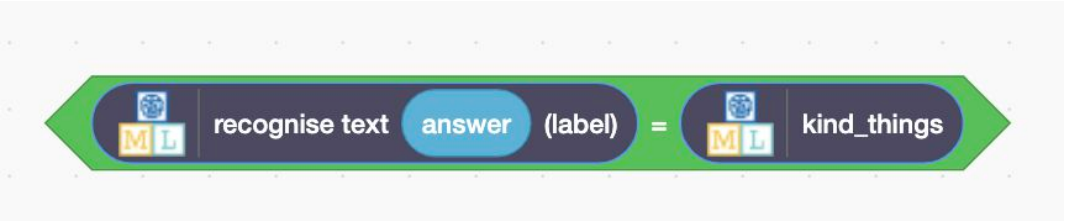
1) When we start the game, display not sure face.
2) Ask the question “Type a message: ”
3) If the answer is recognised as “kind” then, display happy face.
4) If the answer is recognised as “mean” then, display sad face.

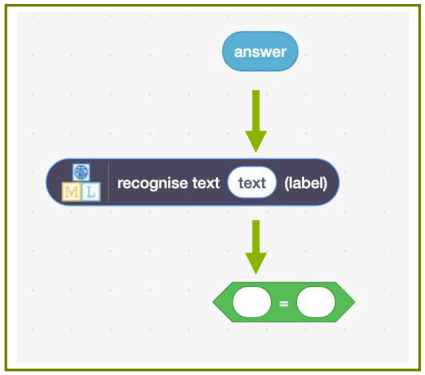
Make Me Happy



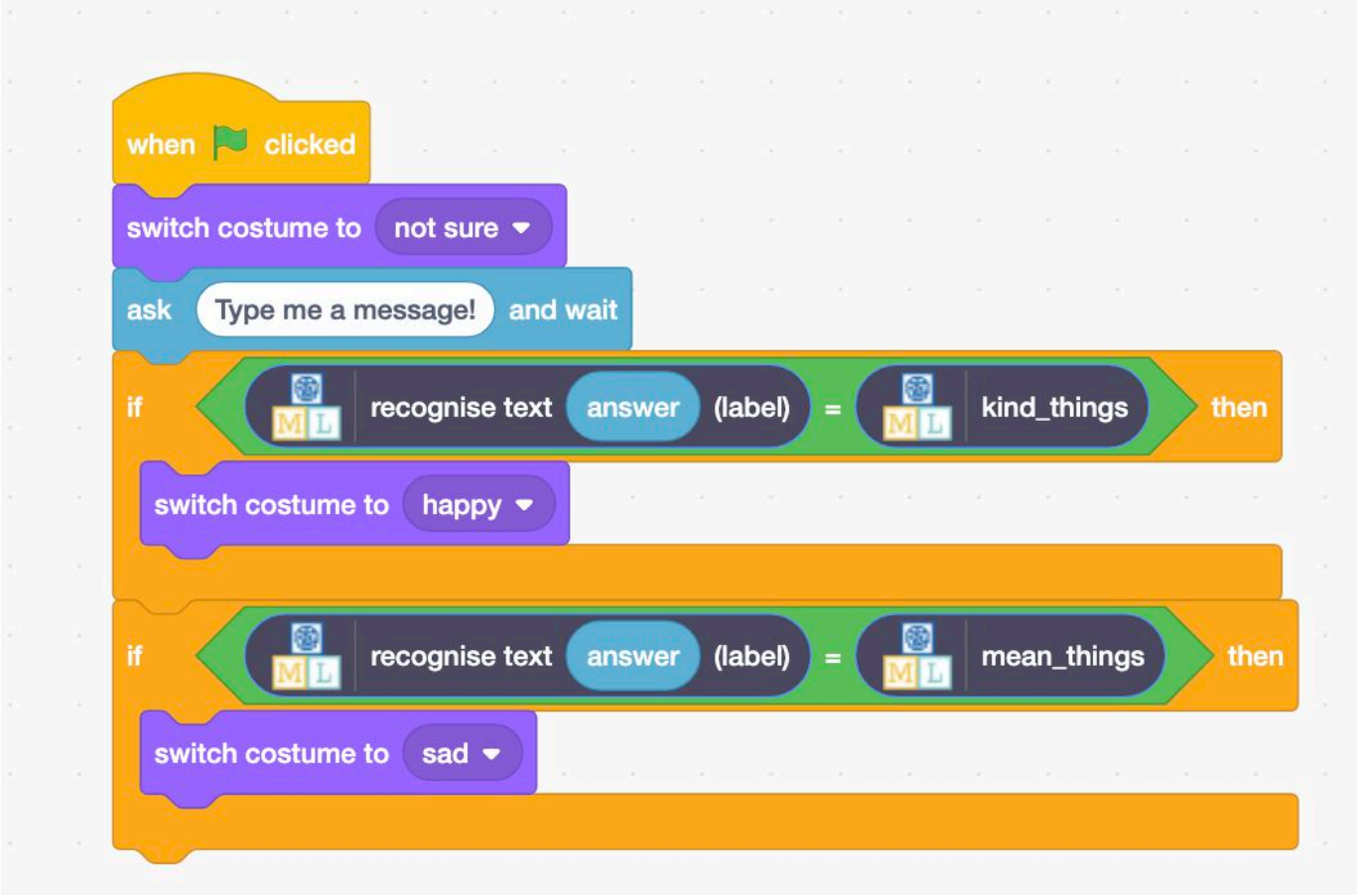








Make Me Happy



Tips

More examples!
The more examples you give it, the better the computer should get at recognising whether a message is kind or mean.

Try and be even
Try and come up with roughly the same number of examples for kind and mean.
If you have a lot of examples for one type, and not the other, the computer might learn that type is more likely, so you'll affect the way that it learns to recognise messages.

Mix things up with your examples
Try to come up with lots of different types of examples.
For example, make sure that you include some long examples and some very short ones.

What have you done?

You have modified your Scratch character to use machine learning instead of the earlier approach.

The more examples you give it, the better it should get at recognising messages correctly.

Activity: 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?

Write a reply (easy)
Instead of just changing the way they look, make your character reply, based on what it recognises in the message!

Try a different character (medium)
Instead of a person's face, why not try something different, like an animal?
It could react in different ways, instead of smiling.
For example, you could make a dog that wags their tail if you say something kind to it!

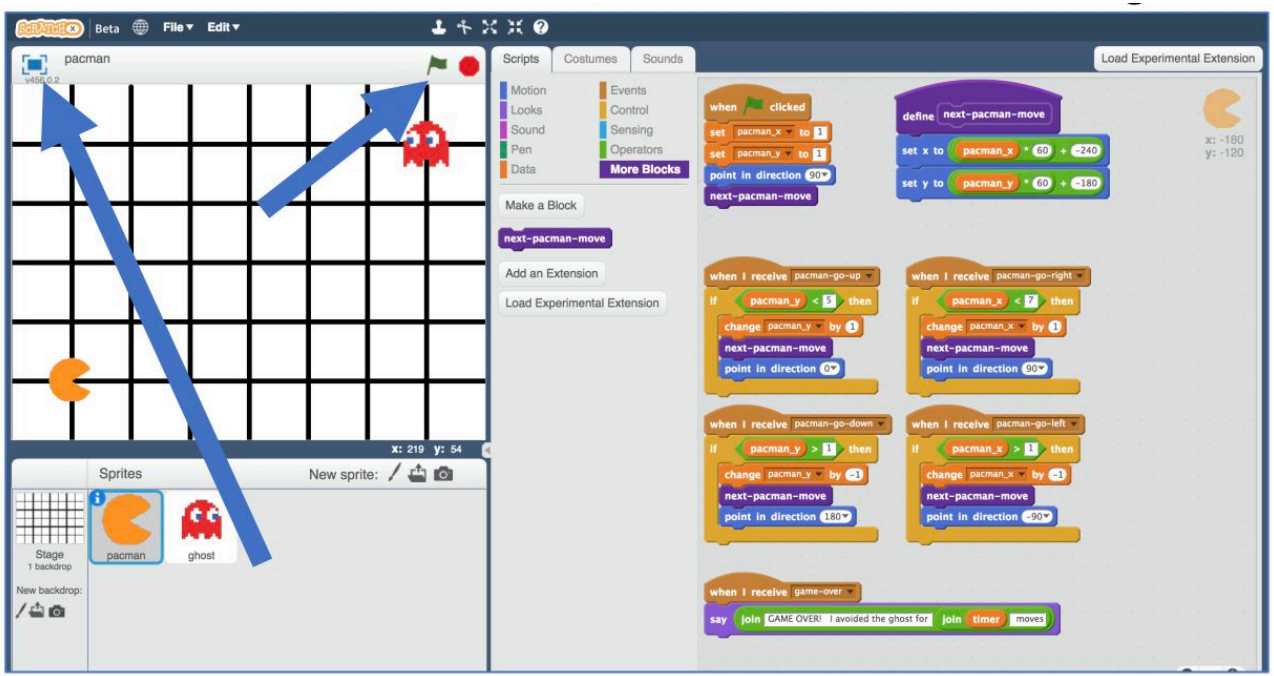
Different emotions (hard)

Instead of kind and mean, could you train the character to recognise other types of message?

Activity: ML Simulation 2

Pac-Man

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



Pac-Man

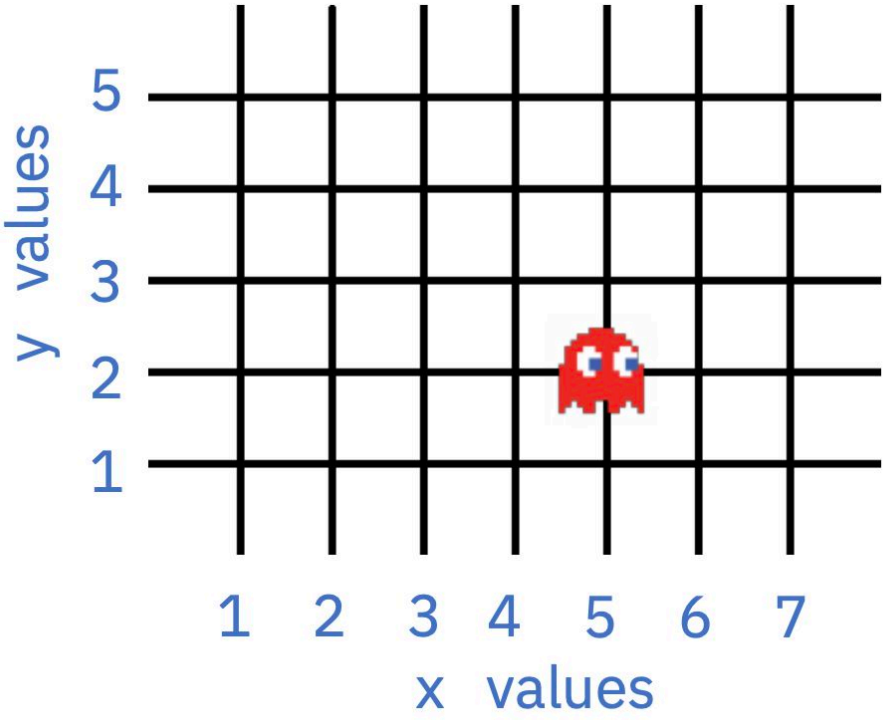
Go to https://machinelearningforkids.co.uk/scratchx in a browser.

Click Project templates -> Pac-Man to open the Pac-Man template for this project.

Play a few games of Pac-Man and avoid the ghost as long as you can.

Pac-Man

Representing Pac-Man in Scratch:



The game board is a graph. Pac-Man and the ghost can only travel along lines.

The location of each character is stored as:

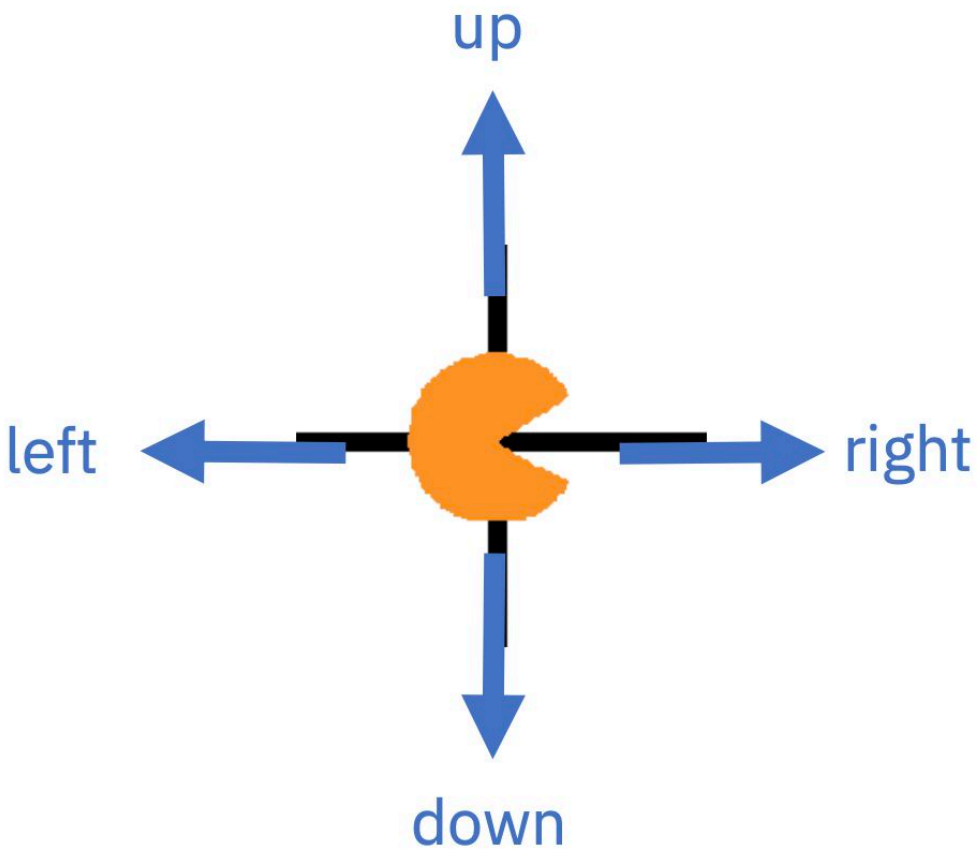
- an x-value (a number from 1 to 7) and,
- a y-value (a number from 1 to 5).

For example, the ghost on the left is at: X = 5 Y = 2.

Pac-Man

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

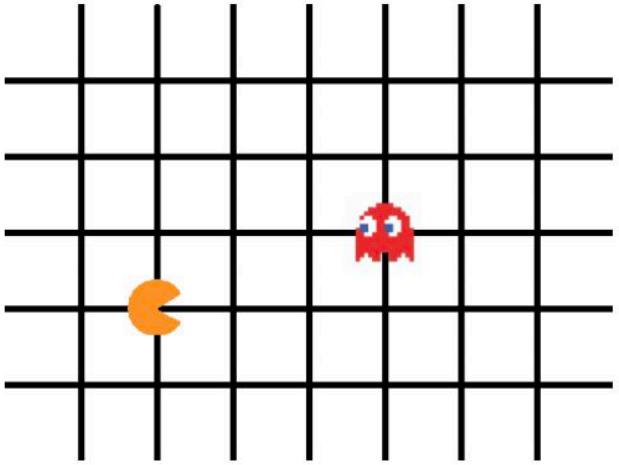
(There are no diagonal moves.)



What are we going to do?

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

Imagine the board looks like this, if you decide to go up:

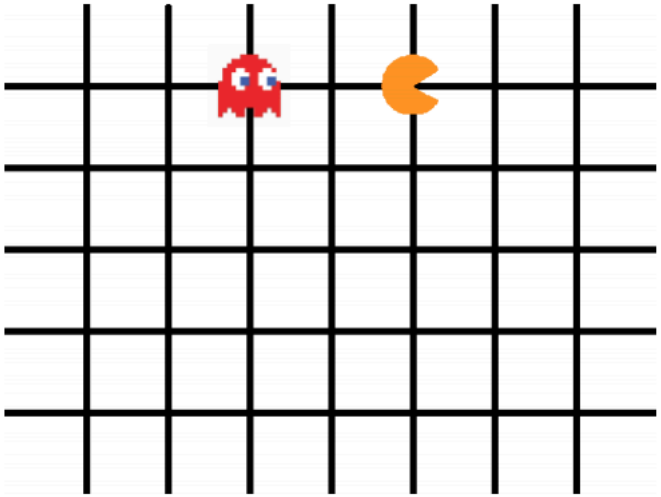


Pacman x	2
Pacman y	2
Ghost x	5
Ghost y	3

This table represents choice "up".

What are we going to do?

Imagine the board looks like this, if you decide to go down:



Pacman x	5
Pacman y	5
Ghost x	2
Ghost y	5

The table represents choice "down".

Pac-Man

Go to “Projects”.

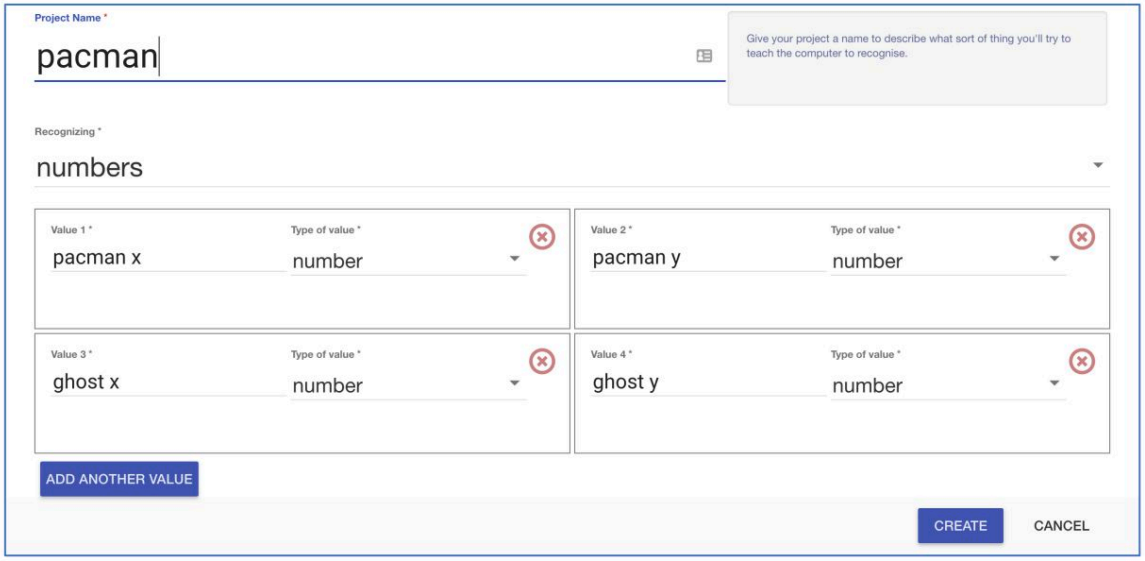
Click the “+Add a new project” button.

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

Pac-Man

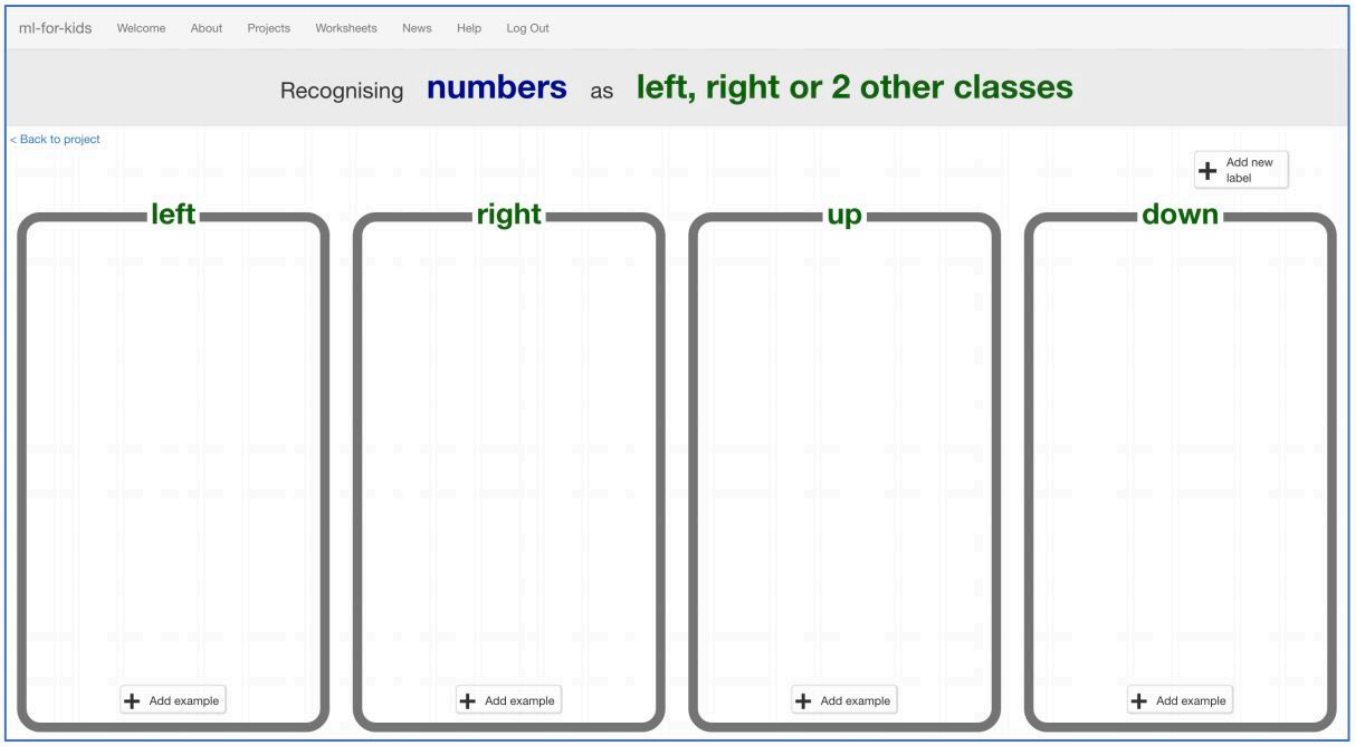
Click “Add a value” and name a value “pacman x” and make it a “number”.

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



Pac-Man

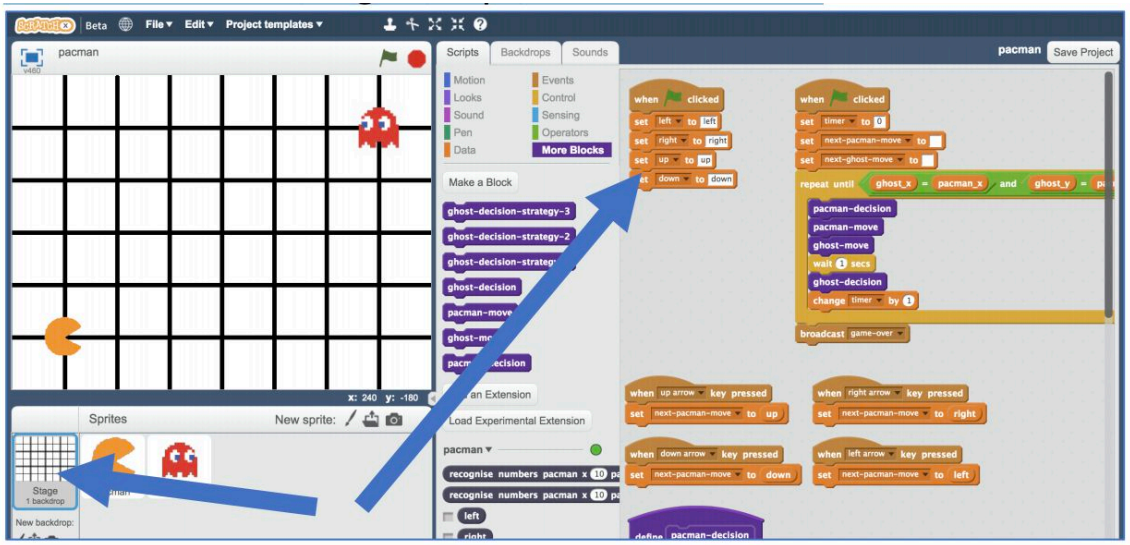
Click "+ Add new label" and create labels shown below:



Pac-Man

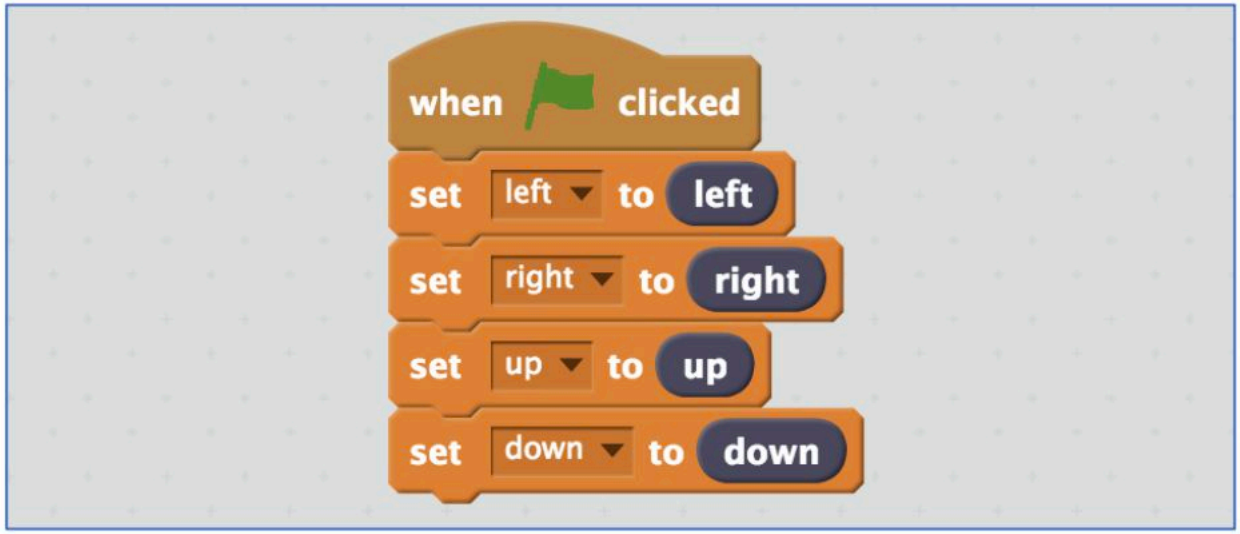
Open the Pac-Man template again.

Click on the “Stage” and find the “when green flag clicked” script that sets the “left”, “right”, “up”, “down” constants.



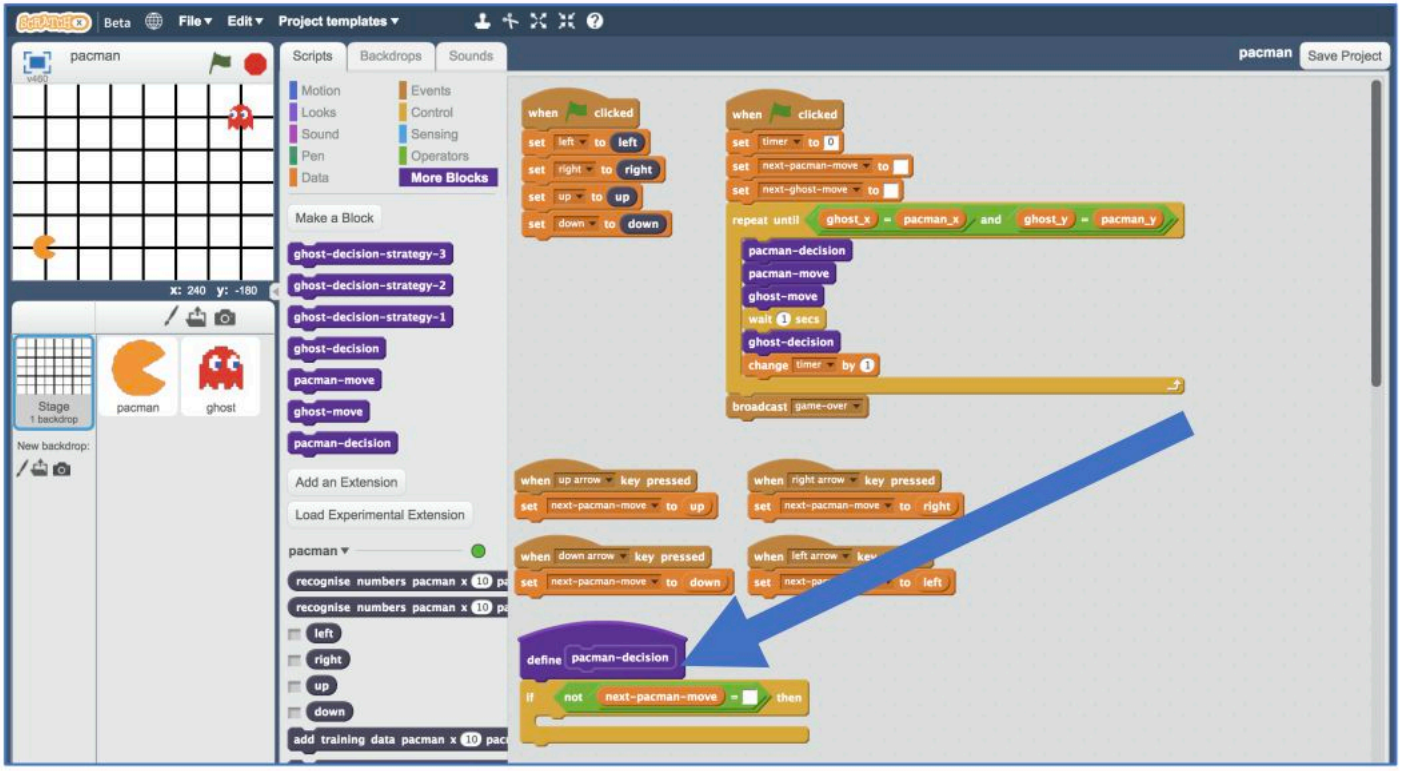
Pac-Man

Modify the script to use your new blocks from the pacman project.



Pac-Man

Find the custom block “pacman-decision”.



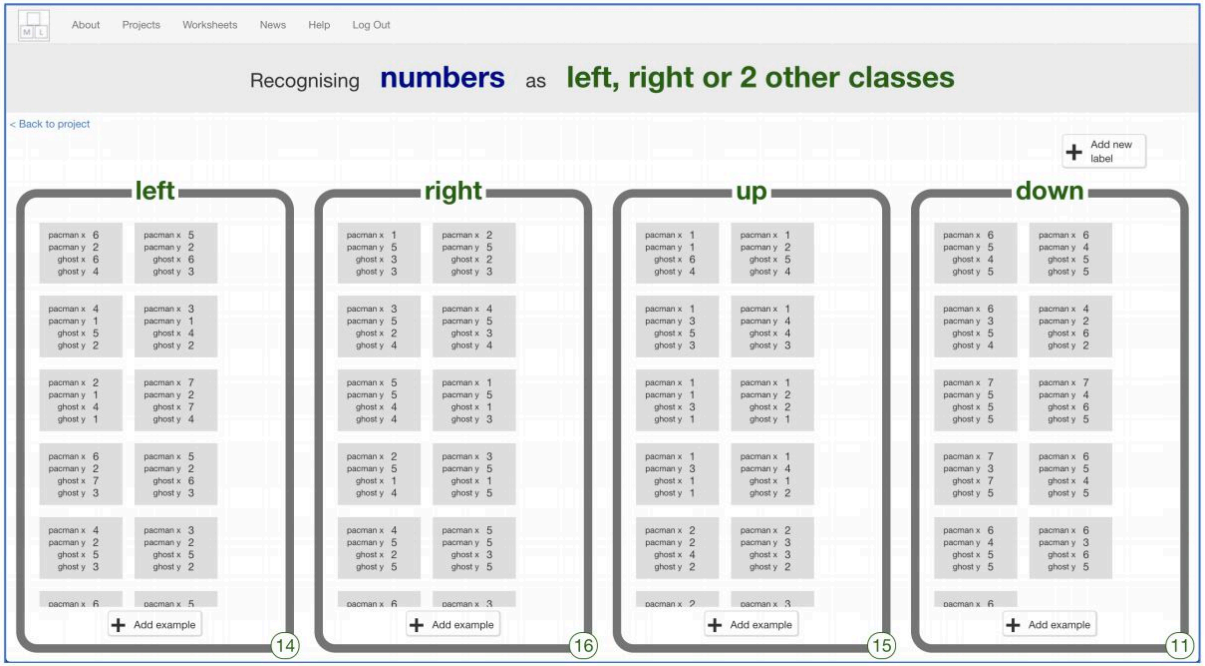
Pac-Man

Update the "pacman-decision" block to add every move you make to your machine-learning training data.

Save your work and call it pacman-learn.

Pac-Man

Click the "< Back to project" link, then click the "Train" button You should see the training examples you collected by playing Pac-Man.



What Have You Done So Far?

You are teaching a computer to play Pac-Man.

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.

You will use those examples to train a machine learning "model".

Pac-Man

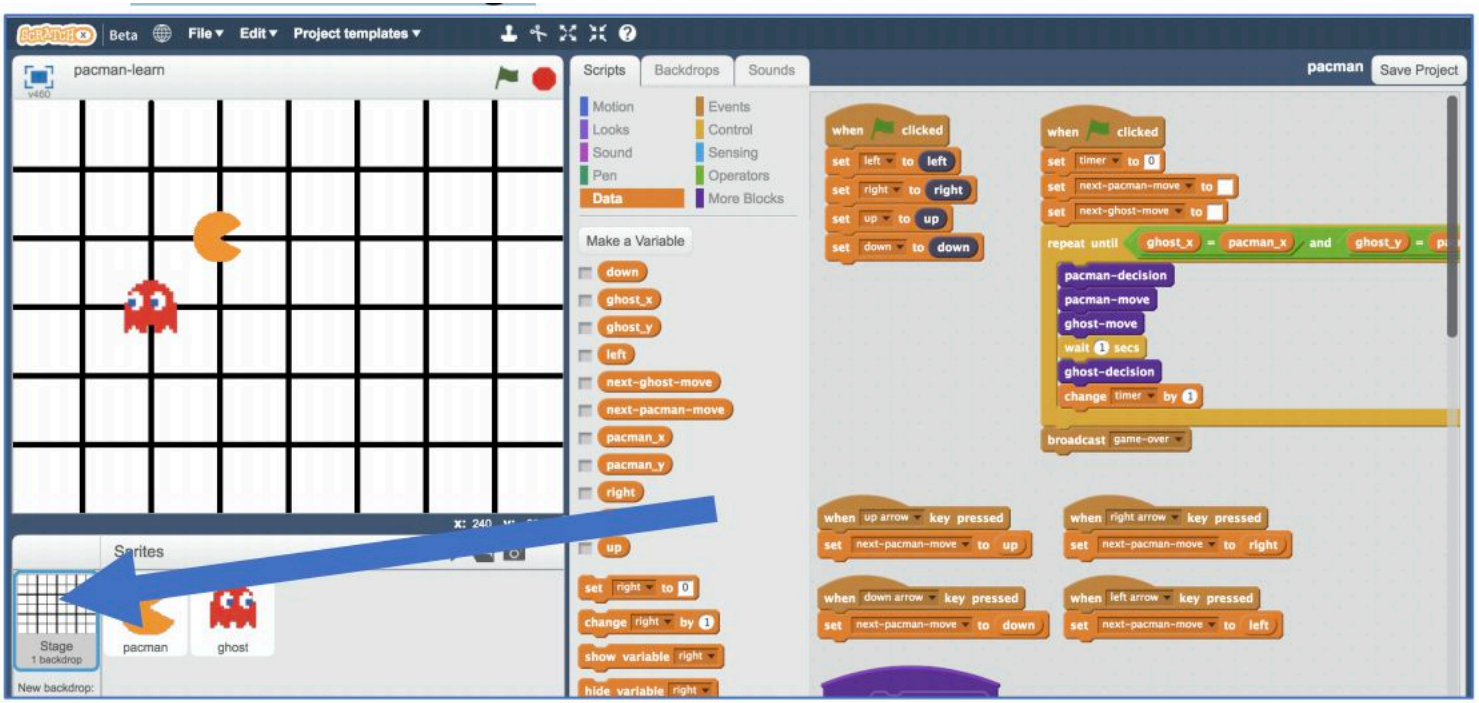
Click the "< Back to project" link.

Click the "Learn & Test" button.

Click the "Train new machine learning model" button.

Pac-Man

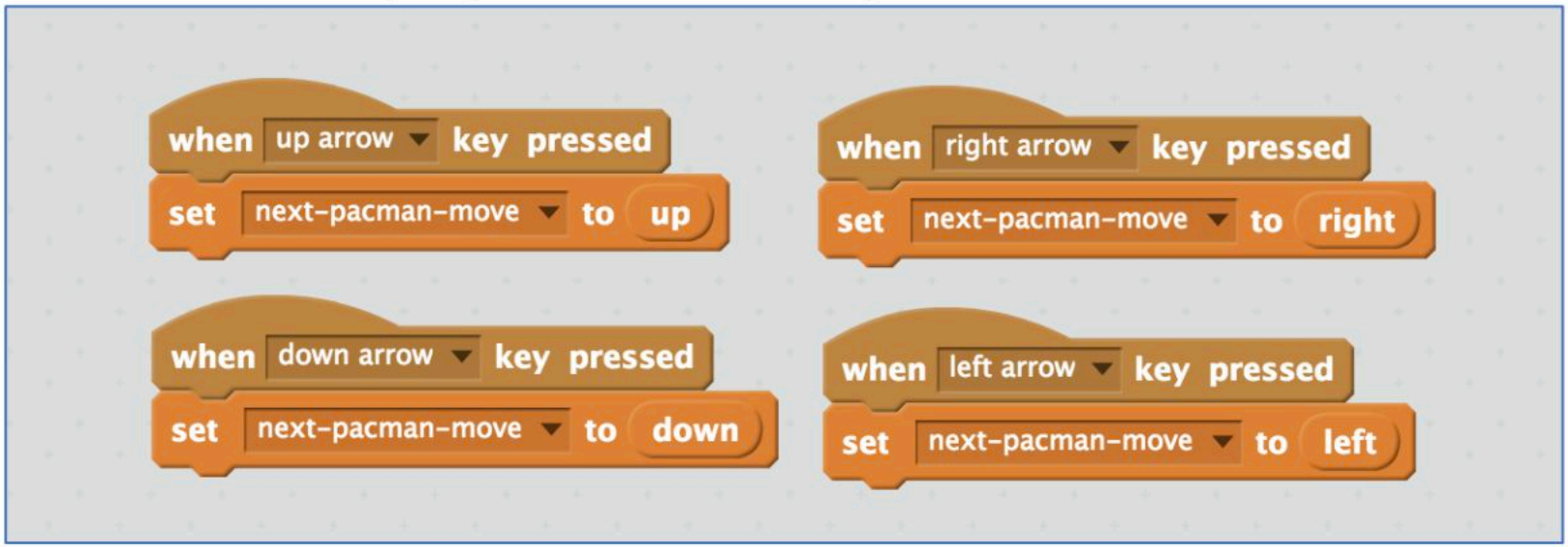
Click on the Stage.



Pac-Man

Delete the key-press scripts because it's the computer's turn.

These are the scripts you do not need any more:

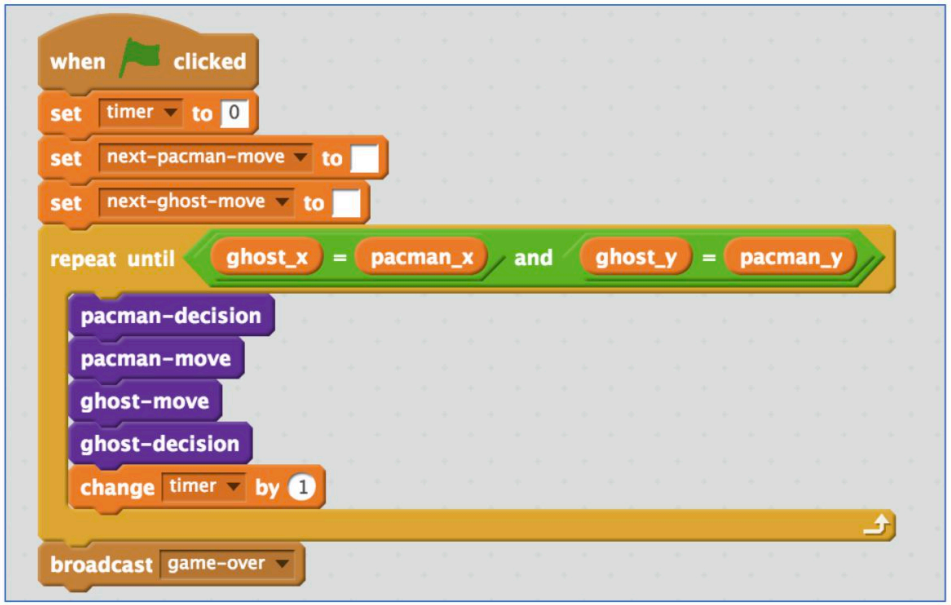


Pac-Man

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

Pac-Man

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



Save your work and call it pacman-play.

How To Improve Your Pac-Man?

Open training project “pacman-learn”.

Play the game even more adding to the training data.

Go back to the training tool. Click the “Train new machine learning model” button.

Open testing project “pacman-play”.

What Have You Done So Far?

You have trained a computer to play Pac-Man. You did not have to describe the rules to the computer.

Instead, you showed it how you play, and taught the computer by collecting examples of decisions that you made when you played.

Tips

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 do not play well, it cannot learn how to play well. If you want it to get better quickly, play as well as you can.

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.

Activity: 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 (easy)

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 (medium)

Try making the game board bigger. Or add obstacles that Pac-Man and the ghost will need to go around.

Make your own game (hard)

This does not only work with Pac-Man. Why not make your own game in Scratch, and then train a machine learning model to be able to play it?

Activity: Create another ML game

Now that you've finished, why not go on to:

https://machinelearningforkids.co.uk/#!/worksheets

And try out the "Smart Classroom" project - *Create a smart assistant in Scratch that lets you control virtual devices*.

Click "Download" -> Quick simplified version of the project, ideal for use as a first introduction to the tool - "Download project worksheet".