

# Ohbot – Using Coding to Control a Physical Device

5<sup>th</sup> October 2022

Total time 1 hour.

Accessible to all levels of familiarity with Ohbot and coding.

## Description

This course introduces the Ohbot app which is a fun and engaging way to learn about coding and to see how coding can be used to control a physical device – a robot head. The Ohbot app includes a simulation so that you can see the results of your coding challenges. No additional items are needed to complete this course. However, participants can work with a physical Ohbot robot if they have one of these available.

## Objectives

Learn about the context of robotics in our lives

Learn coding using the Ohbot App

Learn how to use coding to control a physical device

Review implementation ideas for Ohbot across many various subject areas within K-12

## Preview

5 minutes to read

Q: What is a robot?

A: A robot is any machine that can carry out a series of actions automatically. It doesn't have to be able to move or to have any way of communicating with people. [Wikipedia].

Q: Why do children need to learn about robotics?

A: We are already seeing robots appearing in our day to day lives. ATMs, automated supermarket checkouts and airline check-ins can all be considered to be robots.

All of these robots use electricity to power them, motors and sensors to operate them and computer code to control them. Some of them are frustrating to interact with.

Controlling these machines presents particular challenges of mechanical movement and reliability.

If we accept that these robots are going to be part of our lives we need children and students to learn how to control them and about the specific challenges of controlling physical devices.

Q: How will we communicate with robots in future?

# Ohbot – Using Coding to Control a Physical Device

A: Current robots usually communicate with a touch screen. Science fiction gives us visions of robots that just make beeping noises, others that appear identical to humans and some with human-like robotic faces. There's no need for a robot to communicate with us in the same way as a human but it may be an advantage if we are to work with them successfully. We need children to start imagining how we will communicate with robots and experimenting with different communication methods.

"Children love working with Ohbot and the huge enthusiasm for learning is very apparent in the classroom. Ohbot challenges and engages the children and they develop resilience, as the solution often takes several attempts before they succeed. Ohbot also develops independent thinking and problem solving skills, and provides a positive atmosphere in class." **Clair Harris, Castleview School, Slough, UK**

"Just showed Y6 Ohbot. Whole class gave it a spontaneous round of applause! Can't remember a text book getting that reaction. The students have been really enthused by the work, while at the same time learning about concepts like variables, user inputs, mathematical operators and Booleans. Since using Ohbot, I have noticed the pupils can write and use much more complex programs than before, and seeing the effects of their code on the movements of the robot is really rewarding and engaging." **Peter Johnson, Gosforth Primary Academy, Newcastle, UK**

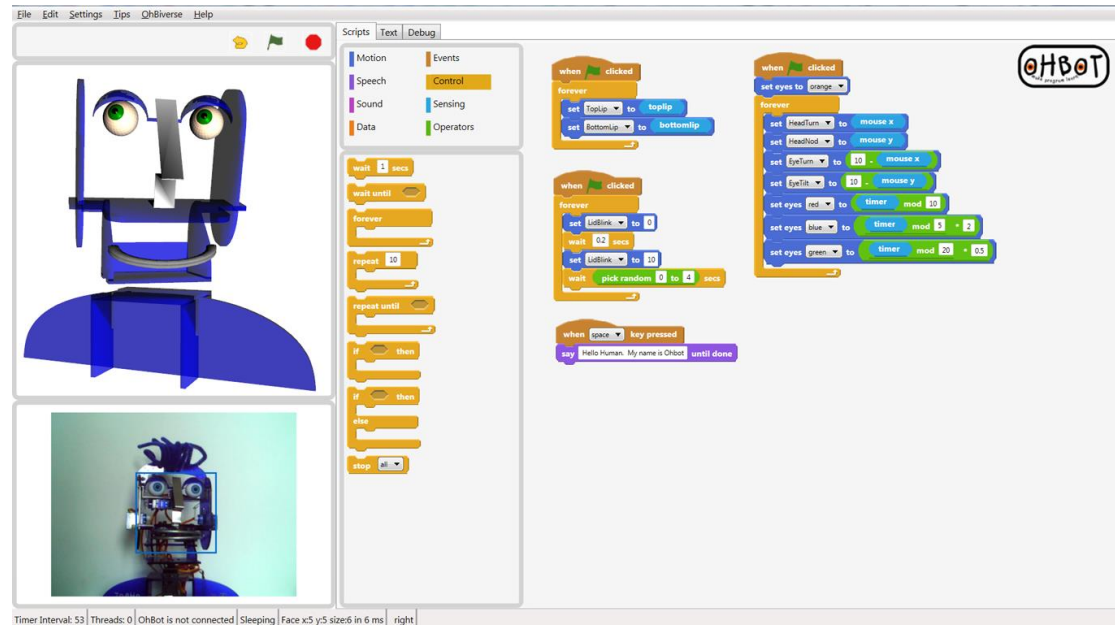
## Professional Journal

- Think of some examples of robots that you already use in your everyday life.
- Think of some ways in which interacting with robots could be improved.
- Think about how you would like to communicate with a robot.

## The Ohbot App

- Easy to install from the Microsoft Store
- Drag and drop block programming interface based on "Scratch"
- Rich language support including variables, loops, conditionals and mathematical operators.
- Models the human face for an engaging experience.
- Full simulation or control of the Ohbot robot through a single cable.
- Full support of text-to-speech and lip synchronisation.
- Face recognition to allow tracking of faces.

# Ohbot – Using Coding to Control a Physical Device



The Ohbot App is an ideal way to get children started with coding or building on skills that they have already acquired using Scratch. As a child's code develops they learn to carry out real world tasks such as storytelling, acting or greeting visitors. Children can easily program movements such as blinking, head and eye movement and lip synchronisation. Ohbot can also detect faces and speak in a range of languages.

## Ohbot in Action



Here are some examples of Ohbot in action:

Ohbot Digital Tales Story Competition - <https://youtu.be/j8ckKbF3fJQ>

Ohbot's journey to Near Space - <https://youtu.be/F8Lz7aW234c>

## Professional Journal

How do you think Ohbot would be received in your classroom?

How would using Ohbot fit with your current coding lessons?

Can you think of an Ohbot challenge that would inspire your class?

## Getting Stared with the Ohbot App

Go the Microsoft Store and search for **Ohbot** or use this link

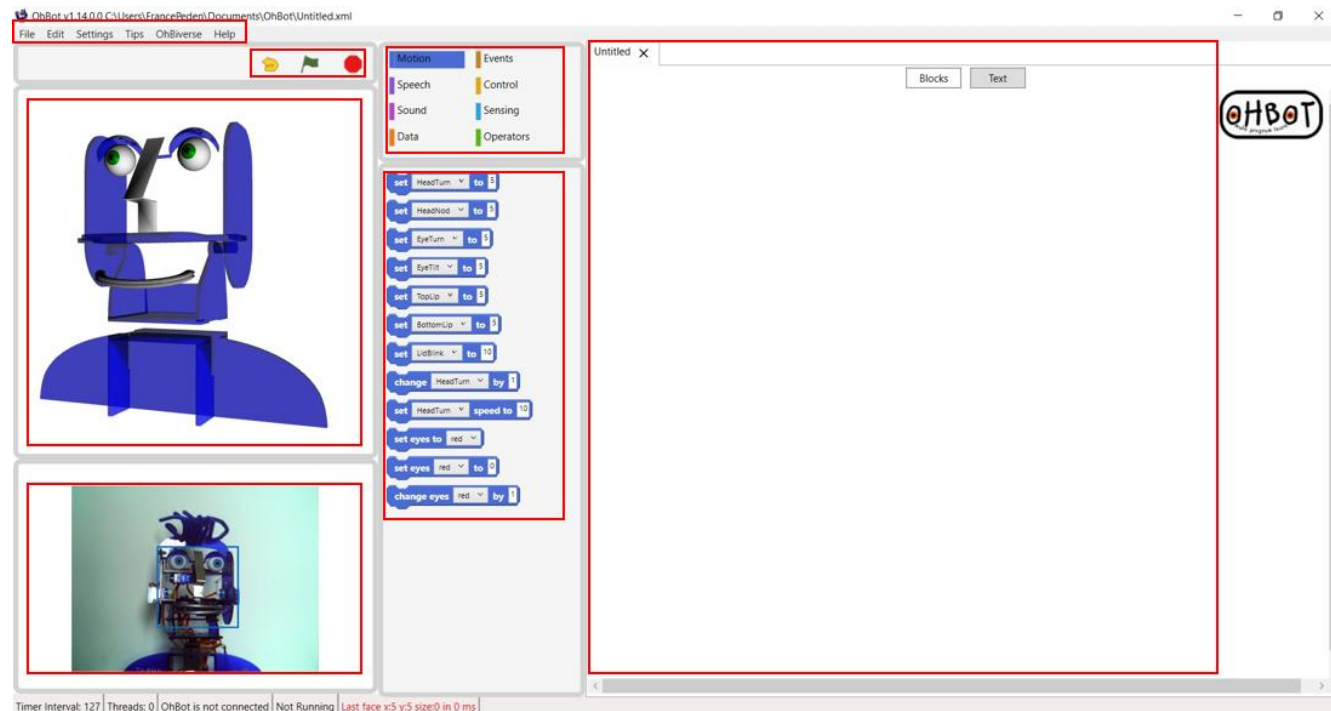
[www.ohbot.co.uk](http://www.ohbot.co.uk)

@OhbotRobot

# Ohbot – Using Coding to Control a Physical Device

<https://www.microsoft.com/en-gb/p/ohbot/9pmnrj2kb3s?cid=install9>

Click on **Get the app**. If a licencing window opens you can enter your email address to register the application. The Ohbot app will open showing this screen:



- **Starting a new file:** click File, and then click New. This gives you a new Scripts area to work in on the right. It is titled Untitled unless you click File, Save As and in the File name box, type a name and then click Save.
- **Saving a file:** File, Save As and navigating to the local drive to save the file.
- **Menu:** File, Edit, Setting, Tips, OhBiverse, and Help.
- **Action buttons:** Beneath the Menu are three action buttons, the **yellow arrow**, the **green flag**, and the **red stop** button. You'll use the green flag to test coding, the red stop button to stop testing before it completes to fix bugs, and the yellow arrow to reset the Ohbot to the normal position before testing code.
- **Ohbot simulation area:** This is where you can test your code by watching what the Ohbot does. If you have a physical Ohbot it will move in a similar way to the simulation.
- **Camera view:** If you have a camera connected this is where you will see the camera image. Detected faces are outlines with a blue box.
- **Block Palette area:** This area contains the Motion, Speech, Sound, Data, Events, Control, Sensing, and Operators options.

## Ohbot – Using Coding to Control a Physical Device

- **Programming block area:** When you select a Block Palette, the different sets of Programming blocks display below. These contain code and will help you make the Ohbot do various things.
- **Scripts area:** You will drag Programming blocks here to write your code and test it.
- **Getting help:** Right click on any programming block and select help to see full help for that block. Click Help on the Help menu for help about the whole app.

### Activity 1 (15 minutes)

#### **Program Ohbot to move and speak using sequences of code.**

Ohbot is a robot that can be programmed to sense the world, 'think' using a computer program and act using motors, speech, sound and coloured light.

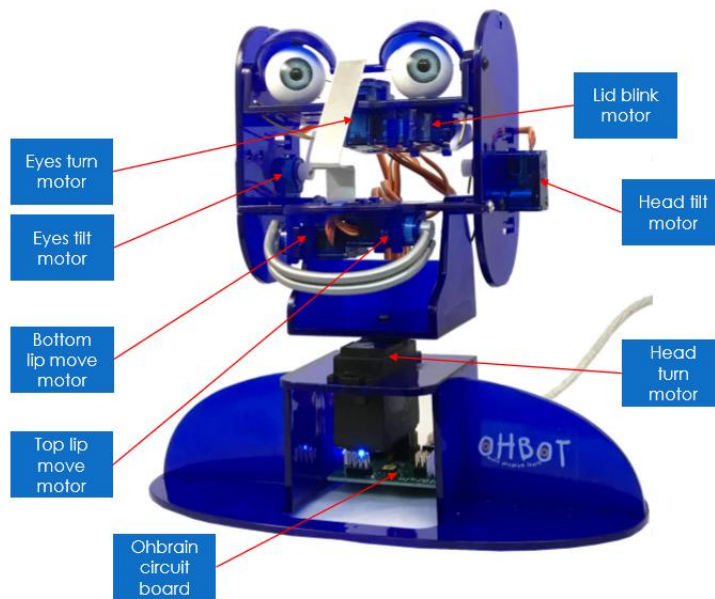
This activity explains how to get started on programming Ohbot's movement and speech.

Ohbots move using servo motors, this type of motor can be instructed to move to particular positions. And by repeatedly setting motors to different positions Ohbot can be made to move continuously, to make it look around or blink for example.

The Ohbot app controls an Ohbot that is attached to the computer's USB port. The app also displays a simulation of the robot which can be used to develop a program but the simulation will not always move in the same way as an actual robot. For example, the simulation moves instantly but the robot can take some time to move between positions.

Ohbot has seven servo motors:

# Ohbot – Using Coding to Control a Physical Device



Ohbot has a reset position, this sets all of its motors to their normal positions:

- Head nod to position 5 (mid point)
- Head turn to position 5 (mid point)
- Eyeball tilt to position 5 (mid point)
- Eyeball turn to position 5 (mid point)
- Eyelids to position 10 (open)
- Top Lip to position 5 (mid point, neither frown or smile)
- Bottom Lip to position 5 (mid point, neither frown or smile)

To reset click the yellow arrow action button at the top left of the programming interface.



## One move

There are several ways to instruct Ohbot's motors. We'll start by making Ohbot move to a particular position when the green flag action button in the program is clicked.



First reset Ohbot to its default position using the yellow arrow.



An event block instructs the computer to start running the sequence, in this case we'll use **when the green flag clicked**, drag it to the scripts area. A motion block set the position of Ohbot's motors. We will use a **set (motor) to (position)** block that defines which servo motor to move and the position it should move to.

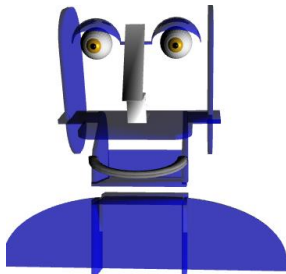


# Ohbot – Using Coding to Control a Physical Device

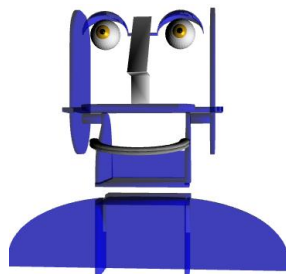


When the green flag is clicked this will set the **HeadNod** servo motor (that tilts Ohbot's head) to **position 0** - Ohbot's head tilted all the way down. This is how different head nod positions will set Ohbot:

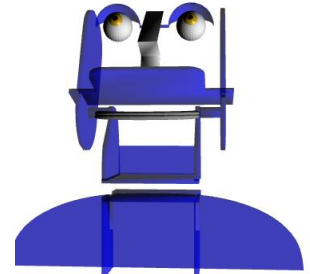
Set HeadNod 0



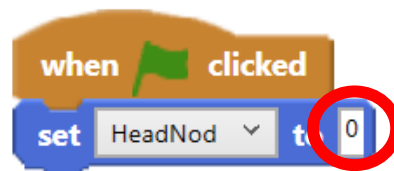
Set HeadNod 5



Set HeadNod 10



Try changing the position value to make Ohbot move to different positions when the green flag is clicked.



Each of Ohbot's motors can be set in the same way.

## Challenge

Try changing your two-instruction program to make it set the position of different motors. Use reset after trying each to see how the code affects Ohbot's movements.

- Look towards the door (using the HeadTurn motor)
- Close eyelids (using the LidBlink motor)
- Turn eyes left (using the EyeTurn motor)

## One action

Ohbot can also speak words, play sounds and if it has illuminated eyes fitted its eyes can be lit in any colour.

# Ohbot – Using Coding to Control a Physical Device

## Challenge

- Make Ohbot give a cheery greeting (using a **say** instruction from **Speech** blocks)
- Play a sound (using a **play sound** instruction from **Sound** blocks)
- Make Ohbot's eyes turn on with a happy colour (using a **set eyes to** instruction from **Motion** blocks)

## Two Movements

A single event such as clicking the green flag can also make Ohbot move more than one motor apparently at the same time. The code below will make Ohbot tilt its head up and close its eyes.



Actually, these run as a sequence. In this case the HeadNod instruction starts first but as soon as it starts it moves on to run the LidBlink instruction. Because there is such a short delay between the first and second blocks running, they appear to run at the same time.

## Challenge

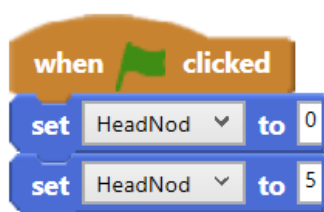
Try making these two actions happen (almost) at the same time

- Head down and eyes closed (HeadNod and LidBlink)
- Smile (TopLip and BottomLip)

## A Sequence of Movements

So far actions have happened as soon as the green flag is clicked. Let's look at how we can program a sequence of movements. Let's make Ohbot nod down and back up again when the green flag is clicked. We need a set HeadNod down instruction and a set HeadNod midpoint instruction to move Ohbot's head.

Can you predict what these instructions will do when the green flag is clicked? Try it?

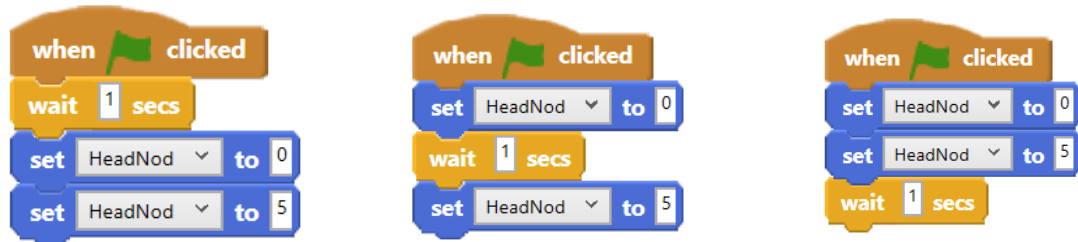




# Ohbot – Using Coding to Control a Physical Device

Did it do what you predicted it would? The program will move Ohbot's head nod straight to position 5, because the HeadNod motor will not have time to move to position 0 before the second instruction runs and sets its position to 5. We need to give Ohbot's motors time to complete one move before instructing it to make another move.

Can you predict which of these sequences will make Ohbot nod?



## Challenge

Code a program that makes Ohbot

- Close its eyelids for one second then open them again
- Shake its head

## A Sequence of Moves and Speech

Let's program Ohbot to say hello, then nod.

There are two main types of say blocks:



Try using each in a separate sequence with a different **when (key) key pressed** event block to start each. One of these will say the text in the say block and as soon as it starts running the instruction move to the next block in the sequence. The other will speak all the text in the say block before running the next instruction.



## Challenge

Code a program that makes Ohbot

# Ohbot – Using Coding to Control a Physical Device

- turn to face you, say a friendly hello to you then turn away.

## Activity 2 (15 minutes)

### Program Ohbot to blink, move its lips and smile when it “sees” you.

This activity explains how to extend your programming to make Ohbot display human like behaviours. In order to do this, we need to use loops, sensors, conditionals and operators.

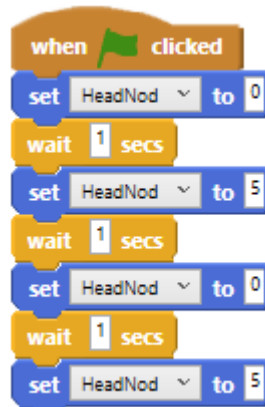
### Repeating Head Nods

We’ve programmed a sequence to make Ohbot nod. It used 4 instructions like this:

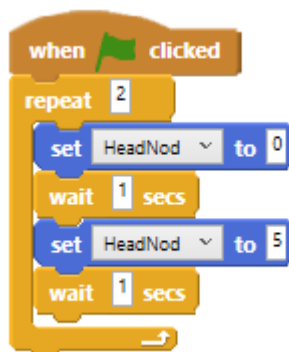


Here are two different ways of programming Ohbot to nod several times:

Here is one way:



And another



# Ohbot – Using Coding to Control a Physical Device

Both programs will do the same thing. But just imagine if we wanted Ohbot to nod five, ten or even more times. The first way would result in a very long program. And in order to make any change to the frequency or length of nods we'd have to make lots of changes to the code. The second way makes it easy to change to the number of times the code is repeated. And to change the length or wait between nods just requires us to change one value for each.

## Challenge

Code a program that makes Ohbot

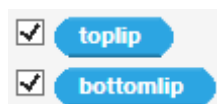
- blink 10 times
- blink in different ways

## Repeating Lip Synch

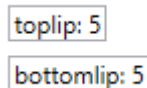
You may have noticed that when a **say** block runs you can hear words through the computer's speaker but Ohbot's lips do not move. To make Ohbot move its lips when it speaks we need to have some code in our program that will instruct the lip motors to move.

One way of doing this is to use a **forever** (repeat) instruction to repeatedly run instructions to set the position of the top and bottom lips to the loudness of Ohbot's speech.

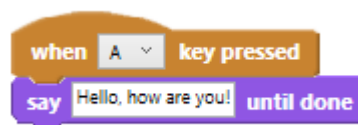
This is available through these blocks found on the sensor menu:



The value of each block depends on the volume of words spoken. To see these values change put a tick into the box next to each. This will display the value of the block in the script area like this:



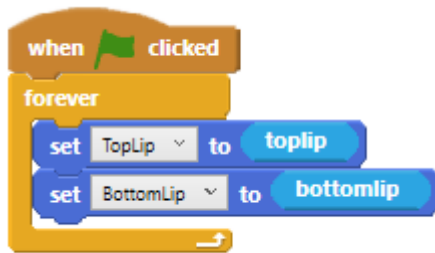
Now run a **say** block. See the value change between 5 and 10 depending on the loudness of the words spoken.



We can use this to synchronise Ohbot's lips with the speech by repeatedly setting the position of the top and bottom lips to the values.

# Ohbot – Using Coding to Control a Physical Device

Try this loop.



## Challenge

Code a program that makes Ohbot

- look around and speak
- set the position of other motors to sensor variables. Try using some of these blocks: **mouse x**, **mouse y**, **sound**
- set random positions for other motors. Try using the **pick random(low) to (high)** instruction from **Operators** blocks

# Ohbot – Using Coding to Control a Physical Device

## Expressions

Ohbot can be programmed to move its lips to smile, frown, look surprised or 'look blank'.

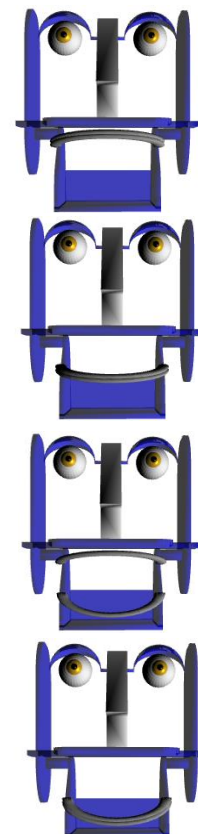
## Challenge

Can you code Ohbot with each of the sequences below. Which expression does each cause? Can you write a program for the missing expression?

```
when A key pressed
  set BottomLip to 10
  set TopLip to 0
```

```
when B key pressed
  set BottomLip to 5
  set TopLip to 5
```

```
when C key pressed
  set TopLip to 10
  set BottomLip to 0
```



Note: The Ohbot software prevents Ohbot's lips crossing over and causing damage to the motors. You may find that this causes them to move in unexpected ways.

## If Then Else Conditional

What if we wanted the program to make a decision about what expression Ohbot showed? Let's try programming Ohbot to smile when a key is pressed and frown when it isn't.

# Ohbot – Using Coding to Control a Physical Device

We can use an **if-then-else** block to work like this:

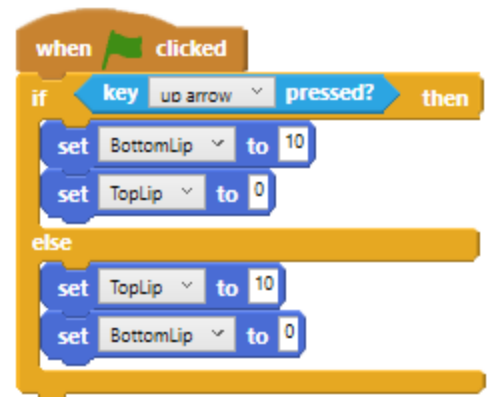
When the green flag is clicked:

If this is true (the up arrow key is pressed):

Do this (make Ohbot smile):

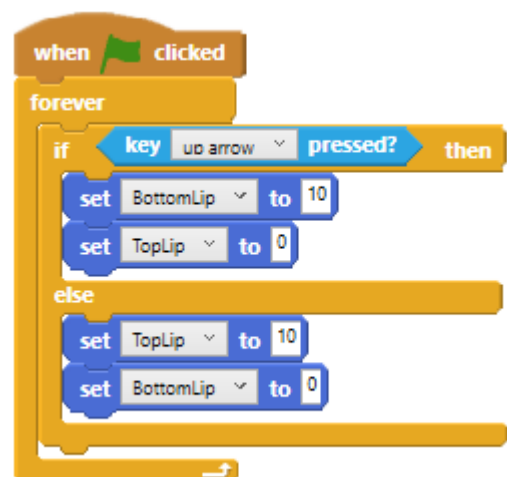
If it is not true (the up arrow key is not pressed):

Do this (make Ohbot frown):



Try the code. You'll notice that it only checks whether the up arrow key has been pressed once (immediately after) the green flag is clicked. It will not take any notice of what happens to the up arrow key after the sequence finishes.

If we wanted to make the program keep checking whether the key has been pressed and running the corresponding code sequence we need to use a **forever** instruction. This will keep repeating the code within it until the program is stopped (using the Red stop button).



Note: You will need a camera connected to your computer for the next section.

What if we wanted to make Ohbot smile when it sensed a face?

There are a number of sensing blocks that return a value based on information from the camera. Try the camera face size block from the sensing menu. Put a tick next to the block to see the value from the sensor block displayed in the scripting area.



size face: 3.4

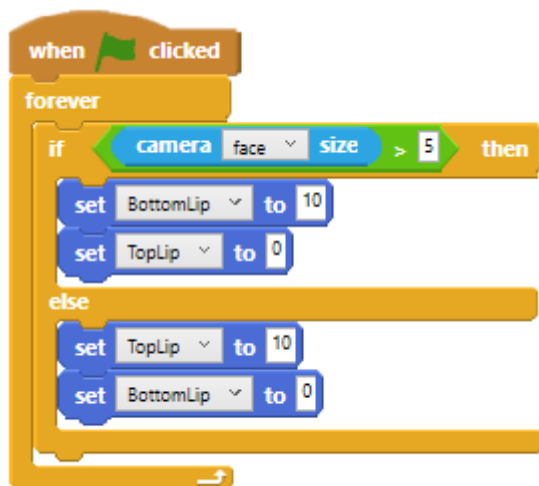
## Ohbot – Using Coding to Control a Physical Device

It gives a value based on how close a face is to the camera. If no face is sensed it gives a value of 0 if a face is really close it will give a value up to 10.

This block cannot be dropped straight into the conditional instruction because it gives a range of values (in this case between 0 and 10). The conditional can only cope with variables that have two states (true or false). However, we can use an operator block to convert the **camera face size** variable into one that gives true or false.



This block has two boxes that can contain a number or another block. The block is true if the value of the left hand box is bigger than the value of the right hand box, otherwise it is false.



### Challenge

Can you code Ohbot

- to say something when it senses a face and something else when it doesn't?
- to smile when it hears a loud noise and frown when it is quiet.

### Professional Journal

Make a note of what you thought of the Ohbot App. Did you feel a sense of achievement when you got the Ohbot simulation to move for the first time? Note down anything that you discovered about the Ohbot App that you will need to support your students when they use it.

### Quiz

10 minutes



# Ohbot – Using Coding to Control a Physical Device

## Question 1

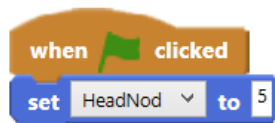
The behaviour of a simulation is always exactly the same as the physical device that it simulates. True or False. [answer is False]

## Question 2

There is only ever one correct way to write a program. True or False. [answer is False].

## Question 3

If Ohbot's head starts in position 5 what will this sequence do:

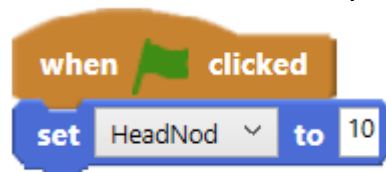


- a) Look up
- b) Look down
- c) It will not move

[answer is c]

## Question 4

If Ohbot's head starts in position 5 what will this sequence do:



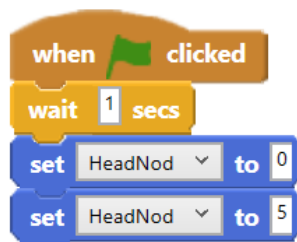
- a) Look up
- b) Look down
- c) It will not move

[answer is a]

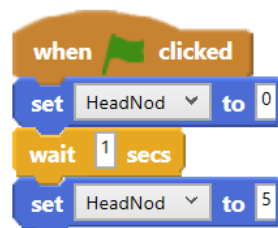
## Question 5

If Ohbot's head starts in position 5 which of these sequences will make Ohbot nod its head:

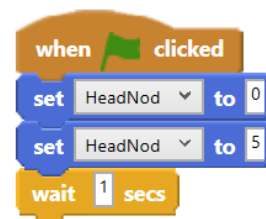
A



B



C



[Answer is b]

# Ohbot – Using Coding to Control a Physical Device

## Question 6

In an Ohbot program a forever loop will stop when

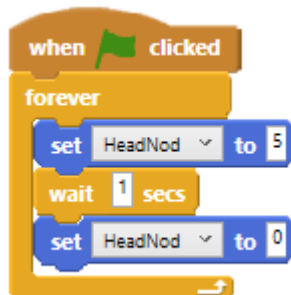
- a) The red stop button is clicked
- b) The sequence is complete
- c) Another sequence starts running

[Answer is a]

## Question 7

If Ohbot's head nod is in position 5

What will this do?

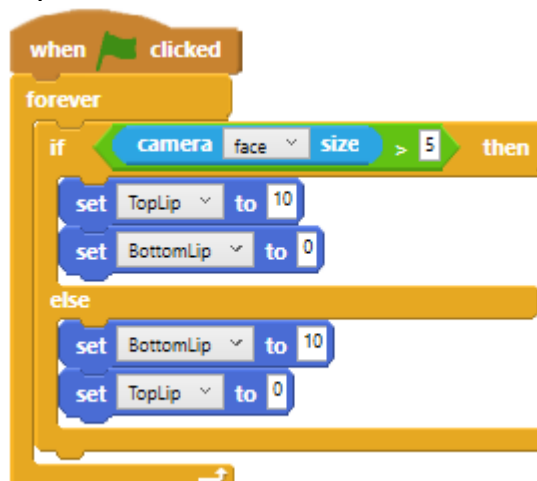


- a) It will not move
- b) It will repeatedly nod up and down waiting for a second in each position
- c) It will nod down once, then stay in that position

[Answer is a]

## Question 8

The space key has been pressed. If camera face size sensed is 3 which sequence will run?



- a) Set TopLip to 10 and set BottomLip to 0
- b) Set TopLip to 5 and set BottomLip to 5
- c) Set BottomLip to 10 and set TopLip to 0

[Answer is c]

## Summary

# Ohbot – Using Coding to Control a Physical Device

5 minutes to read

In this course we have used programming code to control a simulation of a physical device.

We have looked at the specific challenges that have to be considered when programming a machine and we have learned to use the Ohbot app.

We have also experimented with using a robot's "face" as an interface and created code to make a face behave in a human like way.

We have learned about sequences of code, loops, variables, sensors and conditionals.

If you would like to see a physical Ohbot in action there are videos here:  
<https://www.youtube.com/channel/UCplCNFo1O8H6pS53iQIUTFw>

Ohbot can be used across the curriculum for poetry, storytelling, acting and rapping and can speak in any language that's supported by Microsoft Windows.

If you would like to find out more there are resources available for programming Ohbot here:  
<http://www.ohbot.co.uk/code-snippets.html>

There are full projects for you to experiment with here:  
<http://www.ohbot.co.uk/projects.html>

And a scheme of work that can be used with a class here:  
<http://www.ohbot.co.uk/scheme-of-work.html>

Thank you for your time in taking this course.

The Ohbot Team.