## Lesson 23 – Activity Sheet 2

## Setting the Scene

The Activity Sheet will provide you with some starting programs to support you building your micro:PET. They are all based around the speech module and can be adapted and edited to meet the requirements of your solution.

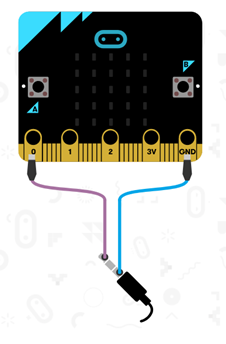
The programs are:

* Hello friend (basic speech)
* Getting too close to your mico:PET (light sensor)
* The micro:PET is hungry (random numbers)
* Feeding your micro:PET

## Getting Started

## **Connect up the Speaker**

Before you start these activities ensure that your micro:bit is wired up to a speaker.



## **Hello Friend (Speech)**

This basic program uses the speech module to enable the micro:PET to talk. Remember that you can edit the settings of the voice to make it sound how you want.

1. pitch – how high or low the voice sounds (0 = high, 255 = low)
2. speed – how quickly the device talks (0 = to fast, 255 = very slow)
3. mouth – how well the words are enunciated (0 = ventriloquist, 255 = Shakespearean actor!)
4. throat – how relaxed or tense the voice is (0 = very tense, 255 = calm)

from microbit import \*

import speech

speech.say("Hello Friend", pitch=64, speed=72, mouth=128, throat=128)

**Getting Too Close to Your micro:PET (Light Sensor)**

The program uses the light sensor to measure the amount of light coming into the micro:bit LEDs. If you stand too close to the micro:PET or you place your hand over the micro:PET’s face, then you block out the light and it responds. You can also adapt the program to make the micro:PET respond to it getting darker and start to fall asleep.

from microbit import \*

import speech

import time

while True:

light = display.read\_light\_level()

#display.scroll(light)

sleep(5)

if light > 50:

dislay.show(Image.HAPPY)

else:

display.show(Image.SAD)

speech.say(give me some space", pitch=64, speed=72, mouth=128, throat=128)

* Line 6 displays the amount of light, which is useful for testing the program is working accurately. Comment it out when done
* Remember to adjust the value of light that the micro:PET responds to on line 8. This will depend on whether you are indoors / outdoors, the lights are on / off or it is sunny or raining

## **The micro:PET is Hungry (Random Numbers)**

This program uses the random module to select a random number between 1 and 50. If the number selected is ‘2’ then the micro:bit displays a sad face. It then uses the speech module to say to you that it is hungry. If the number is not ‘2’ then a happy face is displayed. Use of random numbers and selection are useful for creating random interactions from the micro:PET. For example, if the number 10 is selected then the micro:PET wants attention, if the number 20 is selected then it wants to go for a walk.

from microbit import \*

import speech

import time

import random

while True:

feeding = random.randint(1, 50)

#display.scroll(feeding) #for testing

sleep(1000)

if feeding == 2:

display.show(Image.SAD) #replace with your own image

speech.say("I am hungry ", pitch=64, speed=67, mouth=128, throat=128)

sleep(6000)

elif feeding > 2:

display.show(Image.HAPPY)

sleep(1000)

* Line 6 sets the range of the random numbers between 1 and 50, reduce this to 1 and 4 for testing. This will ensure that you do not have to wait for a long time before the ‘2’ is selected and the micro:PET speaks
* Line 7 displays the random number and is useful for testing the program is working accurately. Comment it out when done
* Line 12 uses sleep(6000) to add a pauses of 6 seconds. You can extend this for the final program, but keep it a short value for testing the program

## **Feeding the micro:PET**

The final example program uses the random module to select a random number between 1 and 50. However, this time if the number selected is ‘2’ the program sets a variable called *hungry* to True and branches off into a sub loop which keeps the micro:PET displaying a sad face and saying to you that it is hungry.

To break out of this sub loop you have to keep pressing Button A, which feeds the micro:PET. Once the PET is fed, it thanks you and the *hungry* variable is set to False. This breaks the sub loop and returns to the original while True loop, restarting the process of selecting a random number. The program starts again.

from microbit import \*

import speech

import time

import random

while True:

hungry = False

feeding = random.randint(1, 4) #change after testing

display.scroll(feeding) #for testing

sleep(1000)

if feeding == 2:

hungry = True

while hungry is True:

display.show(Image.SAD) #replace with your own image

speech.say("I am hungry ", pitch=64, speed=67, mouth=128, throat=128)

sleep(1000)

if button\_a.is\_pressed(): #keep pressing button a to feed the PET

display.show(Image.HAPPY)

speech.say("Thank you for my food ", pitch=64, speed=67, mouth=128, throat=128)

sleep(1000)

speech.say("You are my friend ", pitch=72, speed=67, mouth=135, throat=128)

hungry = False

elif feeding > 2:

display.show(Image.HAPPY)

sleep(1000)

* Line 3 sets the range of the random numbers between 1 and 4, increase this once the program is tested and working correctly. Otherwise you will end up with a very greedy micro:PET that wants feeding every minute!