Micro:pet (V2)

Setting the scene

Loneliness and isolation are a real problem for children staying in hospitals for long periods, especially in rural areas. You have been tasked with creating a digital pet that can be played with and keep people company whilst they stay in hospital.

Success criteria

The product must be suitable for one of the users listed below and the pet must:

- look like a friendly pet (be creative)
- be robust enough to be played with
- contain a micro:bit that users can interact with
- have a face to express emotions when interacted with
- have one or more interactions programmed so it behaves like a pet to keep the user company
- Use the speaker/mic to make your pet talk and react to touch

Some ideas

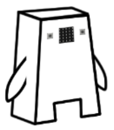
Here are some possible ideas that could be programmed for your pet:

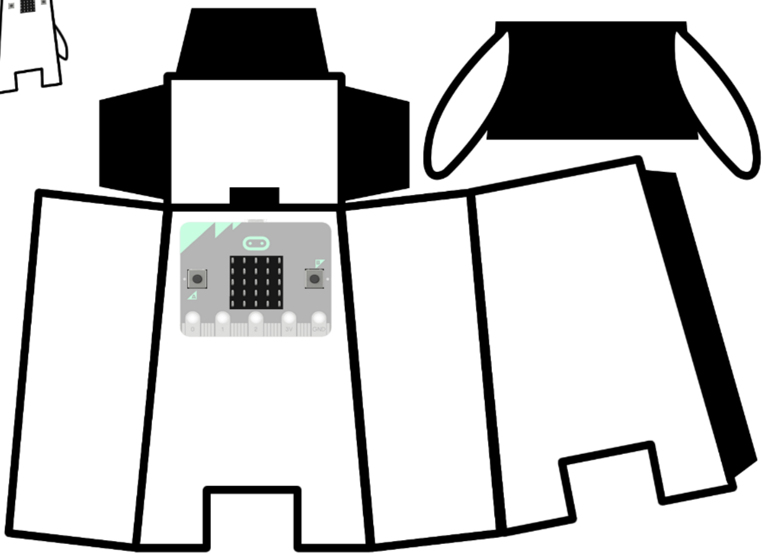
- reacting to playing/shaking (accelerometer)
- feeding (every few hours)
- needing attention (gets lonely if not interacted with frequently) – makes noises to remind you to interact
- sleeping and waking (light sensor) - snoring
- reacting to temperature (temperature sensor)
- mini games (rock, paper, scissors)
- communication/interaction between pets (radio)
- use of other outputs such as sound to make your pet come alive (V2)

Design

You can go one of two ways, using the provided net to use as the body of your pet which you can adapt and decorate or design your own! If you design your own then you will need to complete the design sheet to justify your design ideas.

The most important thing to remember is to be creative and come up with something novel that meets the success criteria in an interesting way. Think about the needs of the user and think about how they will interact with the pet and what they would expect a pet to do and how it would behave.



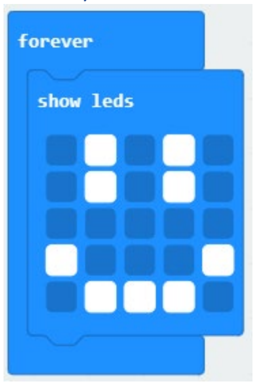


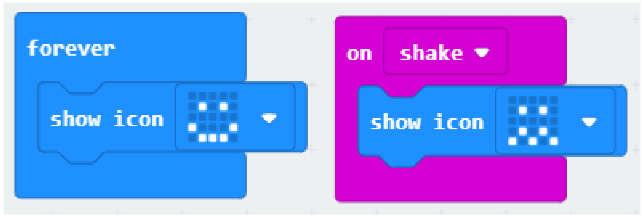
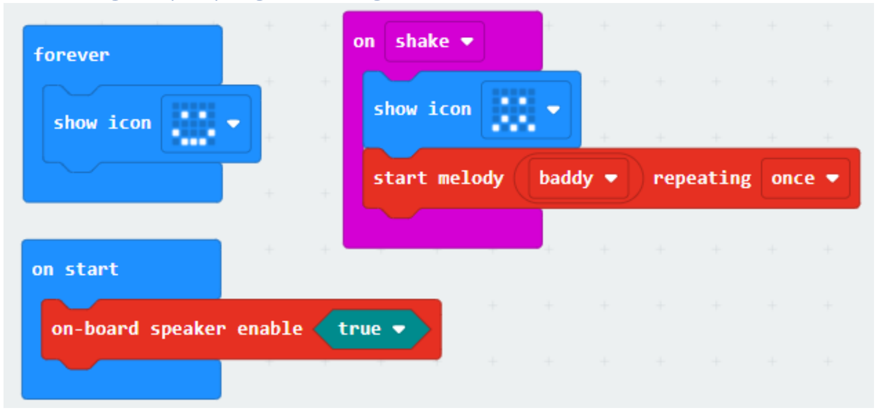
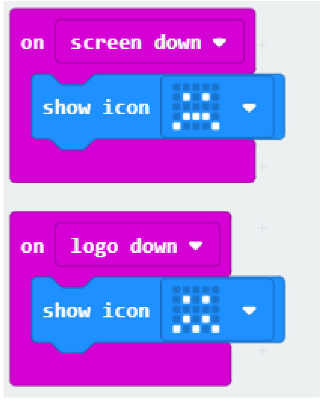
This is a BLANK net. Whilst it would make a minimalist pet you are meant to come up with a name and design for your pet to give it character and to make it come alive for the user.

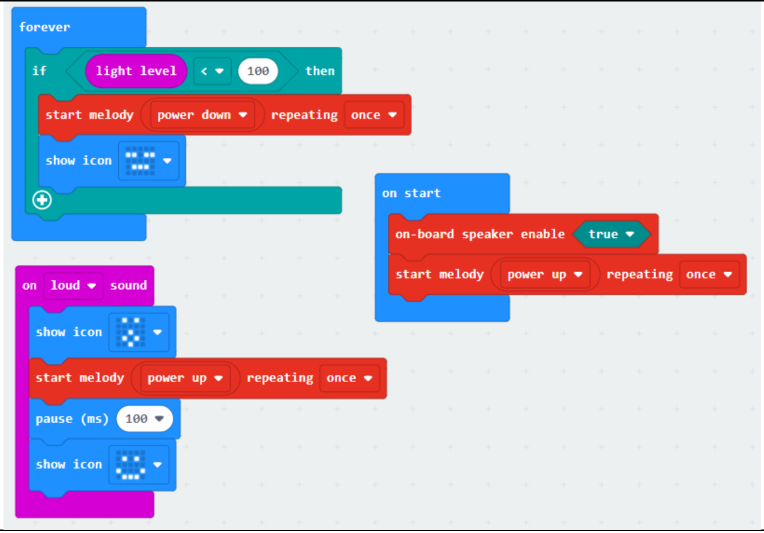
You may also want to design your algorithms. You can do this however you choose or you could jump straight into to makecode. When programming your pet always remember to consider the following:

- How is the user interacting with the pet?
- What are the inputs, processes and outputs?
- Test your pet continuously and adjust and improve as you go along
- Keep in mind the success criteria

Here are some block snippets to help you get started or if you get stuck. Try to do this without using these if you can.

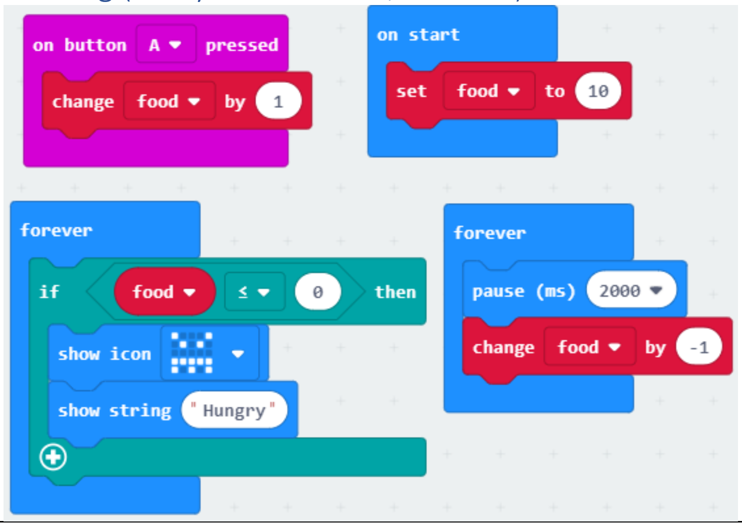
Smiley face for the micro:pet  forever
show leds
`. # . # .`
`. # . # .`
`. . . . .`
`# . . . #`
`. # # # .`
	The first success criteria is to make the micro:pet friendly, we can do this by giving it a smiley face. There are some pre-made icons you could use or you can make your own. Here we add the 'show LED' block into a 'forever' block so the default state is happy.
	
	Pro-tip: Use animation to make your micro:pet more lifelike. Think about how your micro:pet would react and how you can animate this with associated sounds.

Reacting to playing/shaking (accelerometer)  forever / show icon; on shake / show icon	As simple interaction to start with is the micro:pet reacting to being shaken, here we use the micro:bits accelerometer to sense when it is being shaken using the 'on shake' block and then showing a 'sad' face to represent the micro:pet not being happy with being shaken.
Reacting to playing/shaking (accelerometer) (V2)  forever / show icon; on shake / show icon / start melody baddy repeating once; on start / on-board speaker enable true	If you have a V2 micro:bit then you an also add in some sound blocks to accompany your interaction. The V2 sounds are in the 'sound' blocks and you also need to 'enable' the speaker 'on start' to make it work. You only have to enable it once. Think about how sounds can be added to all interactions and what sounds best represent what is happening. Here we use the 'baddy' sound as the micro:pet is not happy but experiment with the sounds available to best suit the interaction.
Emotions that are effected by interaction  on screen down / show icon; on logo down / show icon	Next we will add some other interactions like the micro:pet not being happy about being put screen down using the 'input' blocks. These use the gyroscopic sensor to tell what orientation the micro:bit has and this can be programmed to give outputs such as a sad face. Think about how a micro:pet would react to the available inputs and how the 'reaction' can be moddled using the outputs such as the LEDs and sound.
Wakes up when it hears a loud noise (V2)	If you have a V2 you can expand these interactions with sounds. Here we use the microphone on the micro:bit to sense when a loud noise is 'heard' and give the micro:pet a 'surprised' face and a sound to represent the micro:pet being shocked by a loud noise.



```
forever
    if light level < 100 then
        start melody power down repeating once
        show icon

on start
    on-board speaker enable true
    start melody power up repeating once

on loud sound
    show icon
    start melody power up repeating once
    pause (ms) 100
    show icon
```

Feeding (every few seconds/minutes)

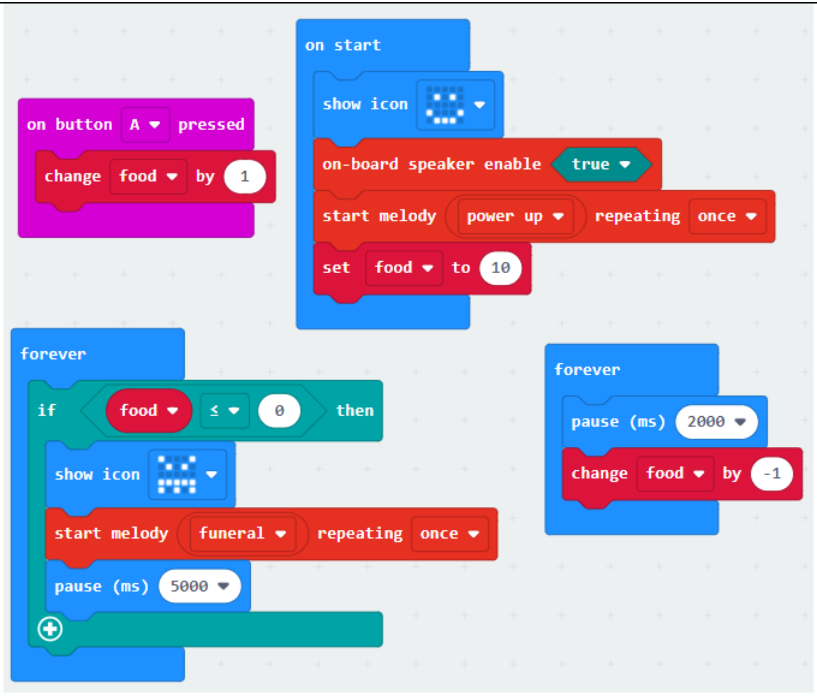


```
on button A pressed
    change food by 1

on start
    set food to 10

forever
    if food ≤ 0 then
        show icon
        show string "Hungry"

forever
    pause (ms) 2000
    change food by -1
```

All pets need regular feeding and the micro:pet is no different. Here we create a variable called food and set it to 10 'on start'. The food variable goes down by -1 every 2 seconds (you can make this as long as you like) and when it gets to 0 the pet gets hungry and lets you know. You can feed the pet by pressing the A button that increments the variable by 1 on each press. Think about how the feeding could be made more interactive. Rather than a button press, another micro:bit could be used as the food and the Radio blocks can be used to recognise when the food is near the micro:pet.

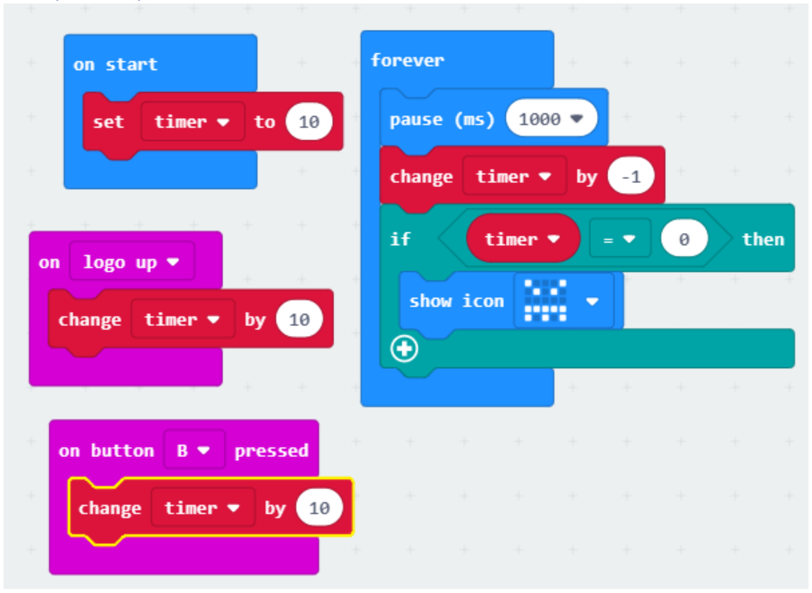
Feeding (every few seconds/minutes) (V2)

If you have a V2 then you can enhance the feeding interaction with sounds for eating, being hungry and even being too full.

```
on button A pressed
    change food by 1

on start
    show icon
    on-board speaker enable true
    start melody power up repeating once
    set food to 10

forever
    if food ≤ 0 then
        show icon
        start melody funeral repeating once
        pause (ms) 5000

forever
    pause (ms) 2000
    change food by -1
```

Needing attention (gets lonely if not interacted with frequently)

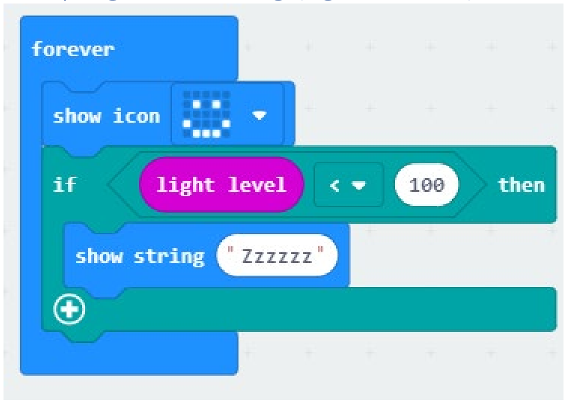


```
on start
    set timer to 10

on logo up
    change timer by 10

on button B pressed
    change timer by 10

forever
    pause (ms) 1000
    change timer by -1
    if timer = 0 then
        show icon
```

Another interaction to explore would be the micro:pet needing attention. This would prompt the user to play with the pet regularly just like a real pet. Here we use another variable called 'timer' that goes down every second and the micro:pet gets angry when the timer reaches 0. The timer variable represents the attention given to the micro:pet. You can give it more attention by pressing the B button or keeping the pet upright which both increase the variable by 10.

Getting the 'neediness' right will take some experimentation as you do not want the pet to become too needy and so becomes annoying to the users.

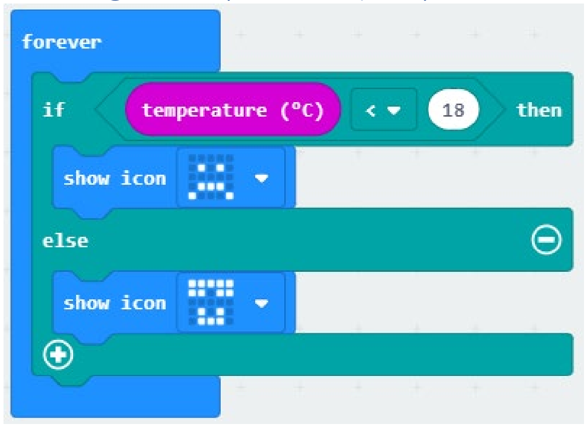
Sleeping and waking (light sensor)



```
forever
    show icon
    if light level < 100 then
        show string "Zzzzzz"
```

Here we also add in an 'if' to measure 'if the light level is less than 100 (out of 255) then show 'zzzzzzz'' this represents the micro:pet going to sleep when it gets dark.

Reacting to temperature (temperature sensor)

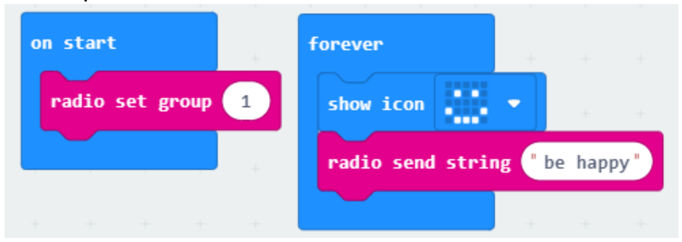


```
forever
    if temperature (°C) < 18 then
        show icon
    else
        show icon
```

Similarly here the micro:pet can be programmed to react to the ambient temperature and get cold and consequently sad.

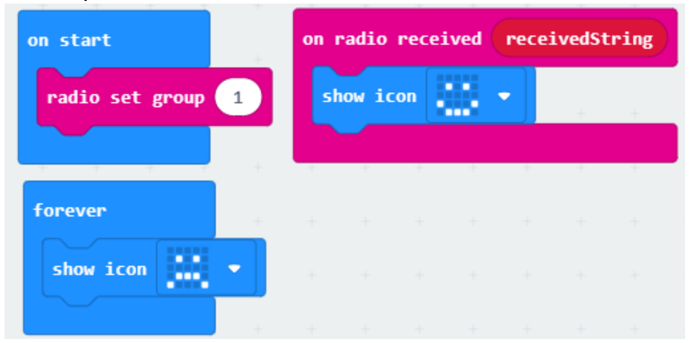
Communication/interaction between micro:pets (advanced)

Micro:pet 1:



```
on start
    radio set group 1

forever
    show icon ▾
    radio send string "be happy"
```

Micro:pet 2:



```
on start
    radio set group 1

on radio received receivedString
    show icon ▾

forever
    show icon ▾
```

If you have more than one micro:bit you can create two micro:pets and have them interact with each other! This opens up many more types of interaction. Here we have an interaction where micro:pet 1 sends a ‘be happy’ string using the radio blocks to micro:pet 2 who when receives the string shows a happy face.

Pro-tip: Both micro:bits need to be on the same radio group to be able to communicate with each other.

The radio blocks can be used to add some advanced features like games between the micro:pets, hide and seek, rock paper scissors as well as adding in more interactive feeding.

Stretch tasks

- Use of other inputs such as other types of sensors (requires additional hardware)
- The medical team have asked whether the pet could log any useful data, investigate what data could be Logged to help the medical team
- Use some servos/motors to make your pet move
- Use https://www.plushpal.app/ to create some custom gestures for your pet

Final thoughts

This project introduces the concept and practice of rapid prototyping a product to meet a need. Mixing together problem solving, making and computational thinking to make creative and innovative projects is what makes physical computing fun. This project could be taken all the way to a real life product and you should think about what other features you could add to better suit the needs of the user. How could it be a better medical device? How could it be better at entertaining a patient? How could it be improved? This idea of iteratively improving products and projects is one we will explore further in other projects.