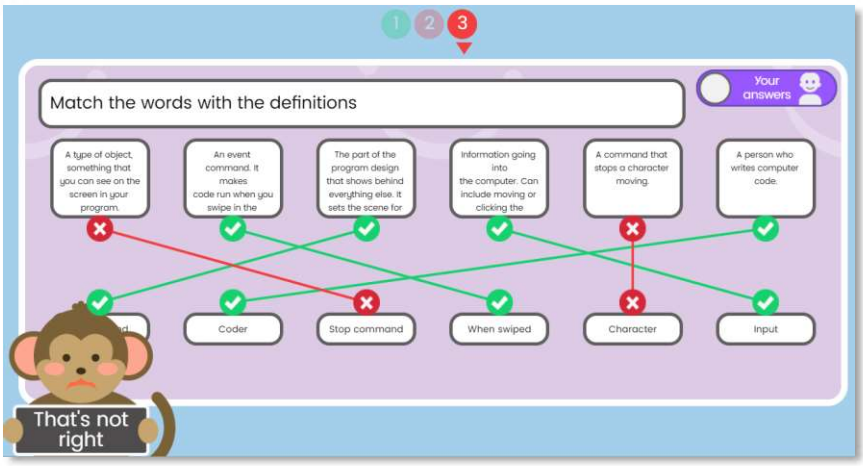
Activities

1. Use the quiz as a class. It is set up so that you attempt all questions and then click the 

button to check the answers. Click 'OK' to see which are correct and incorrect:



You can use the vocabulary cards to find the answers and display in the classroom.

2. Review the word algorithm – how did the pupils show their algorithms in Y1 and 2? They might have used design diagrams, storyboards, or lists.
3. Show the pupils the flowchart example made using 2Chart. Explain that this is another way to show the algorithm for a program.
4. Read the flowchart together. Can the pupils see that there are two objects: an animal and a character? What should the objects do?
   - The animal should make a sound when clicked on.
   - When the background is clicked, the animal should move right, when the animal collides with the character, the animal should stop, and the character should say 'ouch!'.
5. What is the difference between the different coloured boxes?
   - The purple ovals show the start of the sequence, often the initial **input.**
   - The green boxes are **actions** that happen in a sequence.
   - The orange diamond is a **decision**, if the answer is yes, then the yes arrow is followed.
6. Go to the main 2Code Page and show pupils where to find the Free Code Gibbon Icon by scrolling down. Pupils could do this on their own screens at the same time or you could demonstrate.

7. Open Free Code Gibbon on the board. In years 1 and 2, pupils used Chimp level so the options on the screen might look a bit different to the pupils. Review how to add objects in 2Code by going into 

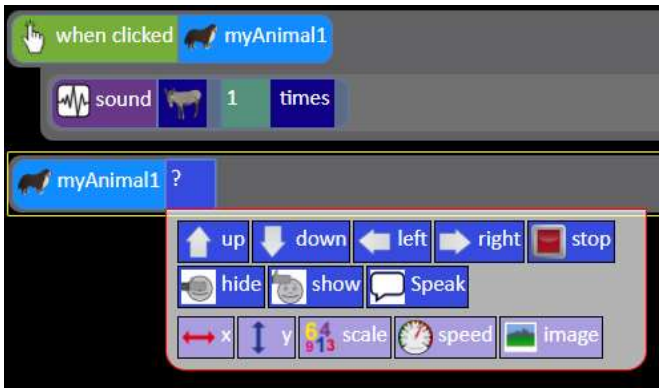
(Design Mode). Drag a car, a character, and an animal onto the background.

8. Return to Code View by clicking 

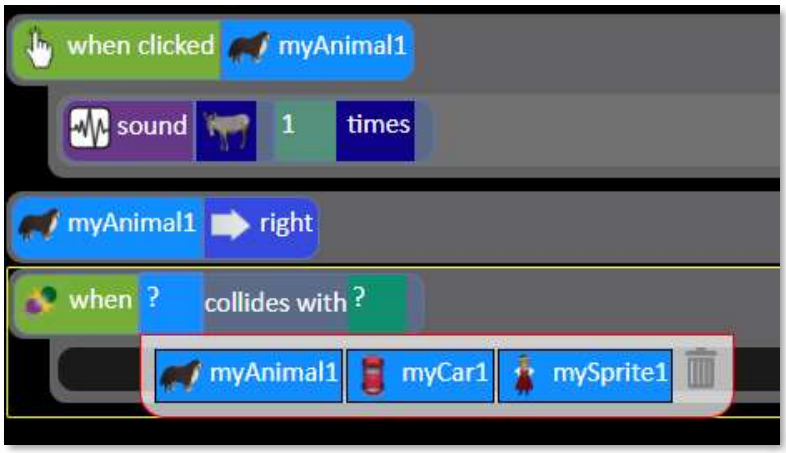
.
9. Remind pupils how to save their work and briefly discuss the need to save regularly so they always have a saved, working version of their program to go back to.
10. Now we are going to try and write code that follows the algorithm.
11. Drag in a When Clicked code block and review how this event works:

12. According to the **algorithm**, the **object** that we want to trigger the **action** is clicking on the animal. Then we need to code what happens when we click on the object. This is called **output**. What should the **output** be?
13. We are going to make the object make a sound when clicked so drag a Sound block into the grey box under When Clicked so that it is indented slightly. Remind pupils that this indentation makes the output happen due to the input (clicking the animal).

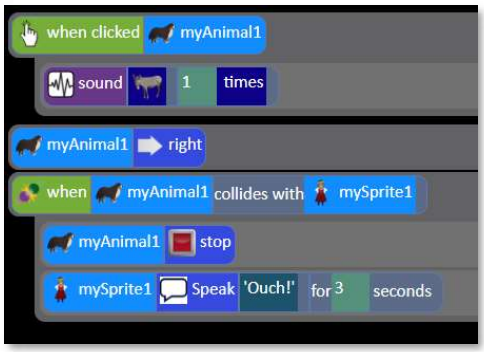
14. Click on the question mark and select a sound.
15. Save and then test the program so far. Does the animal make a sound when you click on it?
16. What else does the algorithm say should happen and how can we make this happen?
17. The animal should move right so drag the animal object into the code window below the existing code.



18. A menu will pop up; ask the pupils what it is called (the **action** menu).
19. Select the action 'right'. The save and test the code again.
20. What do we need to code next? There should be some code when the animal collides with the character.
21. Drag in a collision detection block:



22. Ask the pupils which options to select so that the line or code will do something when the animal collides with the character.
23. What should happen next according to the algorithm? When this happens, the character should say something, and the animal should stop. Ask the pupils to explain how to write the code for this.



24. Save, test and debug (if necessary) the program.
25. What should the car do? Nothing as no action is detailed in the algorithm.
26. Pupils should now be given some time to explore free code Gibbon, reminding themselves of things that they did in years 1 and 2 and trying out new commands that they notice.
27. One thing that they could explore is the different possible actions of the different object types. They could compare the possible actions of the vehicle and the character, for example.
28. If the pupils have workbooks, they can print their code and write about why they chose the commands they did and why they put them in that specific order.