

Digestive System Quiz - Lesson 1

Duration: 45 mins

Overview

In this series of lessons pupils program a quiz to test each others' knowledge of the digestive system. In the first lesson pupils recap the digestive system, design their quiz and explore a Scratch program to learn about commands which will be helpful to them in coding their quiz. In the second lesson they program and debug their code. In the final lesson they evaluate their work, making improvements and reflecting on what they have learnt.

Objectives

Computing

- I can design a quiz
- I can explore code to find useful commands

Science

- I know the simple functions of the basic parts of the digestive system

Before you start and what you need:

- Adapt the [presentation](#) to your school's format as required.
- Ensure pupils have access to the [exploration example file](#)
- Print the [quiz design sheets](#)

Lesson Introduction (5 mins)

- Explain to pupils that over the next three lessons they will designing and programming a quiz to test each others' knowledge of the parts of the digestive system.
- Ask pupils what we might use to create this program? Can pupils recall the steps in programming? *Write an algorithm and use this to help us write the code. Test and debug your code to ensure it is working as expected.*
- Ask pupils to think-pair-share what they can remember about using Scratch from any previous programming. *Guide a class discussion to review pupils' current understanding and to get them thinking about programming in Scratch.*
- Introduce the learning outcomes for the lesson on slide 2, if this is your normal practice.

Main Activity (30 mins)

Quiz design (15 mins)

- Pupils think-pair-share all they can remember about the main parts of the digestive system and their functions. Pupils feedback and note down key points on a whiteboard - if required recap pupils' understanding:
 - http://www.siemens.co.uk/education/pool/teachers/downloads/ks2/the_human_body/scheme_of_work_the_human_body.pdf
 - <http://discoverykids.com/articles/your-digestive-system/>
 - <https://www.pinterest.com/kyliekoch18/digestive-system/>
- Explain to pupils they are now going to design their quiz program. Pupils think-pair-share and feedback what their quiz design will need to show and why, e.g:
 - A sketch of the quiz layout.
 - The algorithm describing the steps or rules in the program.
 - Notes on which Scratch commands they might use to program their quiz and how they will use these.

- If you feel pupils require additional support, show and talk through the example design on slide 3 of the presentation.
- Hand out the quiz design template sheets (either individually or in pairs/small groups depending on your organisation) and give pupils 10 mins to complete their designs.

Mini-plenary idea (5 mins)

- Ask pairs to share their design with another pair. Encourage pupils to evaluate their peers' work, perhaps building a class list of what makes a good design (simple clear sketch of quiz layout, numbers to show the order of steps in the algorithm for example). Give pupils time to improve their work based on feedback.

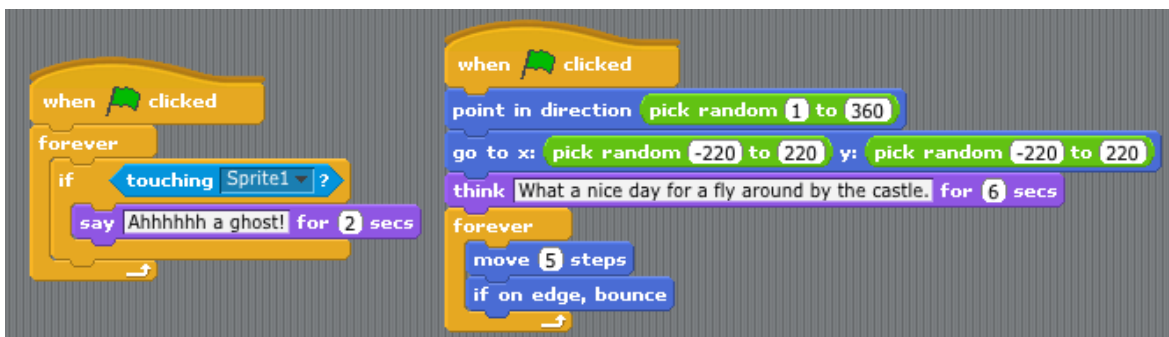
Exploring examples for useful commands (10 mins)

- Open the [exploration example Scratch file](#) on the classroom computer and ask pupils to do the same.
- Show pupils the program is made up of a number of commands (shown below) for two sprites (the programmable objects). Explain to pupils that some of these commands may be helpful when we come to coding our quiz, and that pupils first task is to spend time freely exploring this program to learn as much as they can about the different commands.



```

when green flag clicked
  wait 5 secs
  ask "What's your name?" and wait
  say "join Hello answer" for 2 secs
  forever
    ask "Can you guess how old I am?" and wait
    if answer = 1000
      say "Well done! Yup, I'm 1000 years old today - Happy Birthday to me!" for 5 secs
      stop all
    else
      say "join No I'm not answer" for 4 secs
  
```



```

when green flag clicked
  point in direction pick random 1 to 360
  go to x: pick random -220 to 220 y: pick random -220 to 220
  think "What a nice day for a fly around by the castle." for 6 secs
  forever
    move 5 steps
    if on edge, bounce

when green flag clicked
  forever
    if touching Sprite1?
      say "Ahhhhhh a ghost!" for 2 secs
  
```

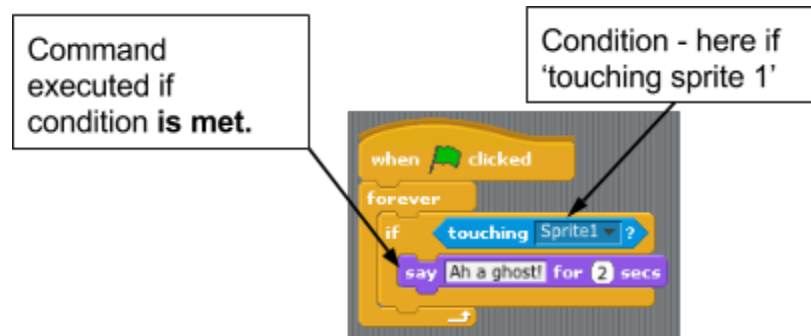
The commands for the two sprites (ghost top; bat bottom) within the program pupils explore.

- Pupils should have 10 mins exploring the commands - running the program, changing bits of code etc. Display slide 4 during this time. Ask pupils to use their **Design Notebook** to note down any useful commands or blocks of commands they find in the example code that they might be able to reuse in their quiz.

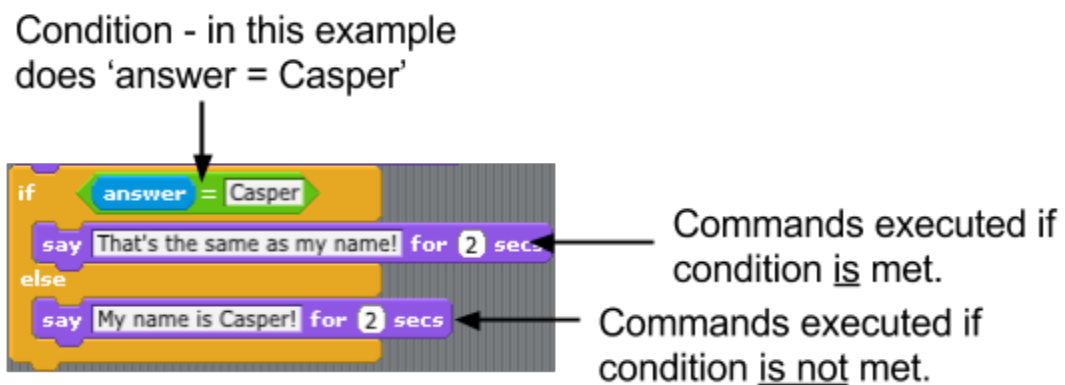
Review of exploration (10 minutes)

- Ask pairs of pupils to compare what they have recorded in their notebook and discuss the useful commands they have found.
- Ask one or two groups to share what they have found, showing the example code to the class using the interactive whiteboard. Discuss as a class how these might be used in their quizzes.

- **Note** - to support pupils in coding their quiz, ensure that the following useful commands have been recognised by pupils and included in their notebook.
 - The **'Ask and wait'** command asks the user the question written into the box and stores the answer in the variable 'Answer'. This can be used to ask questions in our quiz and store the player's answer.
 - The **'If...then'** command is the first example of a **selection**. A condition is placed in the irregular hexagon - here 'Touching sprite 1', then the command within the selection block (Say 'Ah a ghost') is run *if this condition is met*. This command could be used to respond to the person playing our quiz if they give the correct answer.



- The **'If...then...else'** command is a second, more advanced, example of **selection**. A condition is placed in the irregular hexagon - here, does 'Answer = Casper'. The command executes the commands in the top portion if the condition *is* met and executes the commands in the bottom portion if the condition *isn't* met. This could be used so our quiz responds differently depending on whether the player's answer is equal to the correct answer or not.



The 'If...then...else...' selection command being used in the program for tinkering.

Plenary (5 mins)

- Ask two or three pupils/groups to present and talk through their designs. Encourage other pupils to question the groups on their designs and comment on them. What do they like? Why? What do they think they could make even better? Why?
- Guide a brief discussion on how pupils think the Scratch commands in the example program might be useful in coding their designs? What might we use the 'Ask and wait' command for? How might we use a selection command?
- Review the learning outcomes for this lessons from slide 2 of the presentation.

Differentiation

Ideas to challenge pupils:

- Pupils could extend their quiz design to include details of a scoring system.

- Pupils could add extra annotation or notes to their design to improve precision and accuracy.
- Pupils could complete more than one design then choose a final design which might be an amalgamation of their designs, taking the best bits from each.

Ideas to support pupils:

- Pupils requiring additional support could work in a small group with adult support. In the group they could ‘act out’ the quiz program to help write the steps for the algorithm, e.g. *The question is asked, the answer is given, the answer is checked etc.*

Assessment

Emerging	Expected	Exceeding
<p>Requires support to identify important features and create design.</p> <p>Requires support identifying steps in, and writing, the algorithm.</p> <p>Requires support to try out example code or find useful commands.</p>	<p>Identifies important features of quiz and creates design.</p> <p>Writes a correctly ordered algorithm.</p> <p>Runs example code and pinpoints one or two useful commands.</p>	<p>Creates a design with additional detail, such as a scoring system.</p> <p>Writes a precise algorithm with additional detail.</p> <p>Confidently reads code and traces it to the actions it performs and pinpoints most useful commands.</p>

Adapting our planning

This series of lessons can be adapted so pupils’ quizzes test any range of topics they are covering in their curriculum, be it dates from history, cities and countries from geography or number bonds in numeracy.