

Unit 1a Lesson 4: Another planet

40 minutes

Overview

In this lesson children will learn that giving computer instructions in code is called programming, and consolidate understanding that objects can be programmed to do an action when they are clicked on

Learning objectives

Practise coding to make things move when they are clicked

Success Criteria

ALL I can write code to program a rocket to move when it is clicked on

MOST I can write code to program more than one spacecraft to move in different directions when they are clicked on

SOME I can talk about my space scene and explain how I programmed the different space ships to move without bumping into rocks

Key words

program, programming, click/clicked, scene, code, action, object

1. Engage

Prior knowledge:

- Show an example of an app made using Unit 1a Lesson 3 Step 5 and ask children to describe what is happening
- Recap children's understanding of what an **'object'** is (something in an app that does an **action**) and what an **'action'** is (something the **object** does)
- Use the learning wall to help children remember key words: **'object'**, **'action'** and **'clicked'**

New learning:

- Share Learning objectives and Success criteria using the 'I can' statements
- Introduce and add key words **'program'**/**'programmed'**/**'programming'** to the learning wall
- Repeatedly use the key words throughout the rest of the lesson

Key questions

- *Can you describe the **objects** you can see? Can you describe their **actions**?*
- *How are they moving? In which directions?*
- *How is the computer making the **objects** move?*

2. Explore and learn

My Design screen

- Step 1 - Blast off!
- Ask children to identify the **object** and discuss how it would move
- Move the rocket upwards using a mouse and a touch screen, e.g. on the interactive whiteboard

Watch the video

- Step 1 - Blast off!

My Code screen

- Children help to drag the rocket and upwards arrow **code** icons into the click function box
- Explain that the **code** will **program** the rocket to blast off when it is **clicked** on
- Run the app so the children can see the rocket blast off

Challenges

- Set Steps 1-4 as independent or pair challenges

Feedback

- Ask some children to explain what they have done so far - address any misconceptions

Final challenge

- Watch the video for Step 5
- Show children how to use the pen tool to draw some rocks on their space **scene** and erase parts of their design if they make a mistake
- Set Step 5 as a final challenge: ask children to draw rocks into a space **scene**, **program** the spacecraft to move without bumping into them

Key questions

- What is the **object** on this screen?
- What would be a good starting position for the rocket?
- What directions could the rocket move in?
- What instructions could we give the rocket to make it blast upwards?
- How could you make the space shuttle blast the other way?
- Which **code** picture will move the UFO up/down/right/left?
- Can you make the rocket and the UFO move in different directions without them crashing into each other?

3. Evaluate

- Children share their space **scene** app with a partner
- Children explain to each other what they have done using the key word '**program**'
- Children provide two positive items of feedback and one point for improvement

Key questions

- What do you like about your partner's app?
- How could you improve your app?

4. Share

- Show a few examples of children's work to the class
- Revisit the Learning objective and Success criteria
- Ask children to reflect and feedback on how they feel they have done

Key questions

- What did you learn today?
- What was your favourite part of today's lesson?

Learning beyond school

Children visit Espresso Coding at home. They should revisit this lesson and show members of their family how to make space objects move when they are clicked on. They can then challenge a member of their family to create a space scene.

Next steps

Next lesson children will consolidate objectives from Unit 1a Lessons 1-2, they will create a simple program.

Unit 2a Lesson 2: Red Riding Hood

45 minutes

Overview

In this lesson children will learn that a character is an object that can be controlled to do actions when keys are pressed, and begin to develop an understanding of algorithms

Learning objectives

Learn how to code an object to move around the screen when keys are pressed

Success Criteria

ALL I can program Red Riding Hood to move when a key is pressed

MOST I can program Red Riding Hood to be able to move in different directions when different keys are pressed

SOME I can design and program an app that challenges my friend to use different keys to make Red Riding Hood follow a path

Key words

control, character, object, action, key pressed, algorithm

1. Engage

Prior knowledge:

- Show an example of an app that has been created using Unit 2a Lesson 1 Step 6
- Recap children's understanding of concepts learnt so far - refer to the learning wall for key vocabulary
- Work through Step 1 - Press a key, and recap programming using the key pressed event, this time to make objects move instead of disappear

New learning:

- Share the Learning Objective and Success Criteria - using the 'I can' statements
- Identify that in the example just shown, Red Riding Hood is a character that can be **controlled** using the **keys** on the keyboard
- Add key words **character** and **control** to the learning wall

Key questions

- *What is happening in this app?*
- *Can you describe how this app is working using the words **key press** and **action**?*
- *Who is this **character**?*
- *Do you play any games at home with **characters** in?*

2. Explore and learn

My Design screen

- Step 2 - Left and right - watch the video
- Discuss the story of Red Riding Hood and set the gaming scenario of a player using a **keys** to **control** Red Riding Hood
- Ask children to position Red Riding Hood where they would like them to be on the screen

My Code screen

- Ask children help to drag the code icons into the **key pressed** function boxes and explain what will happen when the program is run and the keys are pressed

Challenges

- Steps 2-5 as independent or pair challenges

Feedback

- Ask some children to explain what they have done so far, address any misconceptions
- Watch the video for Step 5
- Introduce a new word - **algorithm**, add it to the learning wall
- Algorithm: a set of instructions| an explanation in words of how we could solve a challenge
- **Challenge:** to make a program to control Red Riding Hood to move in different directions and stop
- **Algorithm:** five parts - we need to make her go left, right, up, down and stop
- **Code:** making the **algorithm** work, write the code for Step 5 with the children:
 - On key pressed= 'a', Red Riding Hood=left, On key pressed= 's', Red Riding Hood=down, On key pressed='d', Red Riding Hood=right, On key pressed='w', Red Riding Hood=up, On key pressed='x', Red Riding Hood=stop

Final challenge

- Set Step 6 as a final challenge: to make a program to control Red Riding Hood to move in different directions and stop (so that she can get to Grandma's house).
- Ask children to write the code that makes the **algorithm** work and solve the challenge

Key questions

- Can you tell me the story of Little Red Riding Hood?
- How could we program Little Red Riding Hood so she can be **controlled** using **keys**?
- Can you explain what you have done so far using the words **character** and **control**?
- What is an **algorithm**?
- Can you say **algorithm**?
- Can you organise the code so that the **algorithm** works?

3. Evaluate

- Children share their apps with a partner/ another group and explain how it works
- Children give each other two pieces of positive feedback and one point for improvement

Key questions

- What do you like about your partner's app?
- How could you improve your app?

4. Share

- Open a shared app and use it as an example to celebrate children's work
- Revisit the Learning Objective and Success Criteria
- Ask children to reflect and feedback on what they feel they have achieved

Key questions

- What have you learnt today?
- Have you achieved the learning objectives?
- How do you feel about your work today?

Learning beyond school

Children visit Espresso Coding at home and show a member of their family how they made Little Red Riding Hood move along a path to get to Grandma's house, and make another app using Step 6 drawing Grandma's house as a finishing point.

Next steps

Next lesson children will learn that events on an iPad can be different to events on a computer, and continue to develop understanding of algorithms.

Unit 3a Lesson 4: Bugs in the garden

50 minutes

Overview

In this lesson children will consolidate their understanding of how code can be programmed to execute at different times. They will practise creating simple animations, using time events to make objects perform actions in a sequence.

Learning objectives

Practise using time to program a sequence of actions and make simple animation

Success Criteria

ALL I can use time in my code to program a spider to move and stop

MOST I can use time in my code to control a snail's movements towards some food

SOME I can explain how I used time in my code to control a snail eating its food without being eaten by birds

Key words

time, sequence, seconds, function box, execute

1. Engage

Prior knowledge:

- Demonstrate an app created using Unit 3a Lesson 3 Step 6
- Recap what the challenge was, what was included in the algorithm and discuss how it works using the words '**after**', '**time**', '**simulation**' and '**sequence**'
- Recap children's understanding of physical systems

New learning:

- Share the Learning objective and Success criteria, using the 'I can' statements

Key questions

- *What is happening in this app?*
- *How is the computer making the lights appear and disappear?*
- *How is **time** used in the code?*
- *What is a physical system?*

2. Explore and learn

Watch the video

- Step 1 - Spiders

Challenges

- Set steps 1-2 as independent or pair challenges

My Design screen

- Show the My Design screen for Step 3. Ask children to explain what they think the challenge is
- Discuss what could be included in an algorithm to solve this challenge
- Ask children to note down how they might write the code to solve this challenge. (Handouts of the three '**after**' **function boxes** could be used to support this.)
- Move the snail and broccoli to different starting positions on the design screen and discuss whether their notes are still relevant

My Code screen

- Write the code with help from the children
- Run the program and watch the timer to see when the snail should change direction
- Correct, debug, run and re-run until the challenge is solved

Challenges

- Set steps 3-4 as independent or pair challenges

Feedback

- Ask some children to explain what they have done so far - address any misconceptions
- Watch the video for Step 5
- **Challenge:** To make a program in which the snail eats the food without getting eaten by the bird
- **Algorithm:** Discuss what would be included in an algorithm for this challenge
- Demonstrate how to scroll down the code screen to see all of the eight 'after' **function boxes**

Final challenge

- Set Step 5 as the final challenge. Ask children to design an app and to write the code that makes the **algorithm** work and solves the challenge
- Move the birds and the food to different starting positions to make it more difficult for the snail to get to the food without touching the birds

Key questions

- *What objects can you see on the screen?*
- *What do we want the snail to do?*
- *What **sequence** of instructions could we use to move the snail to its lunch and then stop to eat it?*
- *How do we know when the snail needs to change direction?*
- *Which piece of code **executes** at what **time**?*
- *Can you tell me what you have done so far using the words '**time**' and '**sequence**'?*
- *How will my code change if I move the objects to different positions at the start of the app?*

3. Evaluate

- Children explain to each other what they have done, using the words '**time**' and '**sequence**'
- Children share their app with a partner and give each other two pieces of positive feedback and one point for improvement

Key questions

- *Did your app work?*
- *Did your **time sequence** work?*
- *Did you have to debug it to fix any bugs?*

4. Share

- Open a shared app and use it as an example to celebrate children's work
- Revisit the Learning objectives and Success criteria
- Ask children to reflect and feedback on what they feel they have achieved

Key questions

- *What did you learn today?*
- *How do you feel about your work today?*

Learning beyond school

Children visit Espresso Coding at home, show a family member their app and explain how they used time to move the snail to the food. Design another app with the objects in different starting positions.

Next steps

Next lesson children will design, write and debug programs that accomplish specific goals. They will use logical reasoning to explain how algorithms work, and detect and correct errors as they work. They will add their own pictures and use time in code to program them.

Sample only