

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT



Subject: Purple Mash Coding

L3 3.1

L4 4.1

L5 4.1

L5 3.1

L6 3.1

L6 4.1

Key Concept/ Theme: To understand what a flowchart is and how flowcharts are used in computer programming. • To understand that there are different types of timers and select the right type for purpose. • To understand how to use the repeat command. • To understand the importance of nesting. • To design and create an interactive scene.

To begin to understand selection in computer programming. • To understand how an IF statement works. • To understand how to use co-ordinates in computer programming. • To understand the 'repeat until' command. • To understand how an IF/ELSE statement works. • To understand what a variable is in programming. • To use a number variable. • To create a playable game.

Prior Learning links:

	Cycle A	Cycle B
Year 1/2	Unit 1.1 Coding • Introducing block coding • Objects and actions • Events (Click event, sound output) • Executing a program • Design view: Planning Unit 1.4 Lego Builders • Algorithms • Logical decision making • Sequencing instructions • Following instructions Unit 2.1 Coding • Algorithms • Collision detection • Timers • Object types • Buttons • Debugging	Unit 1.5 Maze Explorers • Coding a 'turtle' • Creating programs using sequencing and repeat. • Visual use of the Logo programming language. • Program logic and structure. Unit 2.4 Questioning • Logical decision processing. • Forward planning to achieve a solution
Year 3/4	Unit 3.6 Branching databases	Unit 4.5 Logo • Text-based coding • Utilize understanding of coding structures

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

• Logical decision processing • Modelling selection on a binary model

Unit 4.6 Animation

• Sequencing and animation in logical steps.

Vocabulary:

Key Vocabulary

<p>Action The way that objects change when programmed to do so. For example, move or change a property.</p>	<p>Alert This is a type of output. It shows a pop-up of text on the screen.</p>	<p>Algorithm A precise step by step set of instructions used to solve a problem or achieve an objective.</p>	<p>Event An occurrence that causes a block of code to be run. The event could be the result of user action such as the user pressing a key (when Key) or clicking or swiping the screen (when Clicked, when Swiped). In 2Code, the event commands are used to create blocks of code that are run when events happen.</p>	<p>Flowchart A diagram which represents an algorithm.</p>	<p>Implement When a design is turned into a program using coding.</p>
<p>Background In 2Code the background is an image in the design that does not change.</p>	<p>Bug A problem in a computer program that stops it working the way it was designed.</p>	<p>Button A type of object that responds to being clicked on.</p>	<p>Nesting When coding commands are put inside other commands. These commands only run when the outer command runs.</p>	<p>Input Information going into the computer. Can include moving or clicking the mouse, using the keyboard, swiping and tilting the device.</p>	<p>Interval In a timer, this is the length of time between the timer code running and the next time it runs e.g. every 1 second.</p>
<p>Click Event An event that is triggered when the user clicks on an object.</p>	<p>Code Writing the code for a computer program.</p>	<p>Collision Detection Event The event of two objects colliding.</p>	<p>Properties These determine the look and size of an object. Each object has properties such as the image, scale and position of the object.</p>	<p>Object Items in a program that can be given instructions to move or change in some way (action). In 2Code Gibbon, these include character, turtle, button, vehicle, animal, food, shape, number, input and label.</p>	<p>Predict Use your understanding of a situation to say what will happen in the future or will be a consequence of something.</p>
<p>Command A single instruction in a computer program.</p>	<p>Debug/Debugging Fixing code that has errors so that the code will run the way it was designed to.</p>		<p>Sequence When a computer program runs commands in order.</p>	<p>Repeat This command can be used to make a block of commands run a set number of times or forever.</p>	<p>Run Clicking the Play button to make the computer respond to the code.</p>
			<p>Turtle Object A type of object in 2Code that moves by coding angles of rotation and distance to move.</p>	<p>Test To run the code and observe what happens to identify where there might be bugs in the program.</p>	<p>Scene In 2Code, this is the combination of the background and objects in a program.</p>
					<p>Timer Use this command to run a block of commands after a timed delay or at regular intervals.</p>

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

Key Vocabulary

Key Vocabulary

Action
 The way that objects change when programmed to do so. For example, move or change a property.

Background
 In 2Code the background is an image in the design that does not change.

Command
 A single instruction in 2Code.

Execute
 This is the proper word for when you run the code. We say, 'the program (or code) executes.'

Alert
 This is a type of output. It shows a pop up of text on the screen.

Button
 A type of object that responds to being clicked on.

Debug/Debugging
 Fixing code that has errors so that the code will run the way it was designed to.

Algorithm
 A precise, step-by-step set of instructions used to solve a problem or achieve an objective.

Code blocks
 A way to write code using blocks which each have an object or an action. Each group of blocks will run when a specific condition is met or when an event occurs.

Design
 In coding, this is a plan for the program showing the visual look of the user interface (the screen) with the objects. The algorithm can be represented as part of the design, showing actions and events.

Event
 An occurrence that causes a block of code to be run. The event could be the result of user action such as the user pressing a key (when Key) or clicking or swiping the screen (when Clicked, when Swiped). In 2Code, the event commands are used to create blocks of code that are run when events happen.

Nest
 When coding commands are put inside other commands. These commands only run when the outer command runs.

Implement
 When a design is turned into a program using coding.

Repeat until
 In 2Code this command will repeat a block of commands until a condition is met.

Flowchart
 A diagram that uses specifically shaped, labelled boxes and arrows to represent an algorithm as a diagram.

'If/Else' Statement
 A conditional command. This tests a statement. If the condition is true, then the commands inside the 'if block' will be run. If the condition is not met, then the commands inside the 'else block' are run.

Object
 Items in a program that can be given instructions to move or change in some way (action). In 2Code Gibbon, these include character, turtle, button, vehicle, animal, food, shape, number, input and label.

Predict
 Use your understanding of a situation to say what will happen in the future or will be a consequence of something.

'If' Statement
 A computer uses an IF statement to decide which bit of code to run. If a condition is true, then the commands inside the block will be run.

Input
 Information going into the computer. This could be the user moving or clicking the mouse, or the user entering characters on the keyboard. On tablets there are other forms such as finger swipes, touch gestures and tilting the device.

Prompt
 A question or request asked in coding to obtain information from the user in order to select which code to run.

Repeat
 This command can be used to make a block of commands run a set number of times or forever.

Run
 Clicking the Play button to make the computer respond to the code.

Key Vocabulary

Properties
 These determine the look and size of an object. Each object has properties such as the image, scale and position of the object.

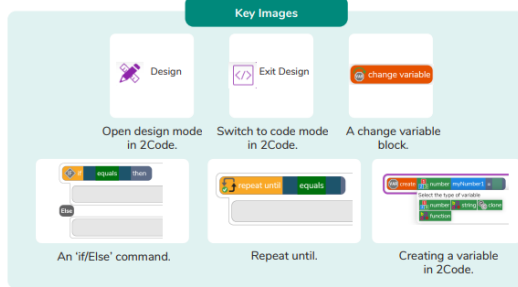
Timer
 In coding, use a timer command to run a block of commands after a timed delay or at regular intervals.

Selection
 Selection is a decision command. When selection is used, a program will choose which bit of code to run depending on a condition.

Sequence
 This is when a computer program runs commands in order.

Variable
 A named area in computer memory. A variable has a name and a value. The program can change this variable value. Variables are used in programming to keep track of things that can change while a program is running.

Key Images



Key Images:

Resources needed for each lesson – 2dos to set.

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

Unless otherwise stated, all resources can be found on the main unit 3.1 page. From here, click on the icon to set a resource as a 2do for your class. Use the links below to preview the resources; right-click on the link and 'open in new tab' so you do not lose this page.

- Free Code Chimp (this is found on the main 2Code page). Preparation
- Set Free Code Chimp as a 2Do, call it 'Repeat Command'. You can select the following objectives when setting the 2Dos to make future assessment easier:

Year:	<input type="text" value="Y3"/>	▼	
Subject:	<input type="text" value="Computing"/>	▼	
Strand:	<input type="text" value="Computer Science"/>	▼	
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.			<input checked="" type="checkbox"/>
Use sequence, selection and repetition in programs; work with variables and various forms of input and output.			<input checked="" type="checkbox"/>
Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.			<input checked="" type="checkbox"/>

L4 4.1

Resources Unless otherwise stated, all resources can be found on the main unit 4.1 page. From here, click on the icon to set a resource as a 2Do for your class. Use the links below to preview the resources; right-click on the link and 'open in new tab' so you do not lose this page.

- Is It Raining 2Code Example.
- Is It Raining IF Flowchart.
- Is it Raining IF ELSE Flowchart.
- Reginald Rocket 2Code Example.
- Reginald Rocket IF ELSE Flowchart.
- Free Code Gibbon (this is found on the main 2Code page).
- Storyboard Planner. Preparation • Set Free Code Gibbon as a 2Do.
- Set Reginald Rocket 2Code Example as a 2Do. You can select the following objectives when setting the 2Dos to make future assessment easier:

Year:	<input type="text" value="Y4"/>	▼	
Subject:	<input type="text" value="Computing"/>	▼	
Strand:	<input type="text" value="Computer Science"/>	▼	
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.			<input checked="" type="checkbox"/>
Use sequence, selection and repetition in programs; work with variables and various forms of input and output.			<input checked="" type="checkbox"/>
Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.			<input checked="" type="checkbox"/>

- Print copies of the Storyboard Planner for children to use if you are using it (see step 7)

L5 4.1

Resources Unless otherwise stated, all resources can be found on the main unit 4.1 page. From here, click on the icon to set a resource as a 2do for your class. Use the links below to preview the resources; right-click on the link and 'open in new tab' so you do not lose this page.

- Free Code Gibbon This is found on the main 2Code page.

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

- Genie.
- Night and Day.
- Variable Game Cards.
 - 2 copies of Number Cards 0-23.
 - 2 boxes (transparent, if possible, e.g. ice-cream or take away containers). Preparation
- Set Genie as a 2Do. You can select the following objectives when setting the 2Dos to make future assessment easier:

Year:	Y4	▼
Subject:	Computing	▼
Strand:	Computer Science	▼
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.		<input checked="" type="checkbox"/>
Use sequence, selection and repetition in programs; work with variables and various forms of input and output.		<input checked="" type="checkbox"/>
Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.		<input checked="" type="checkbox"/>

- Set Night and Day as a 2Do (if using extension)
- Print and cut up 2 copies of number cards 0-23.
- Print and stick 4 Variable Game Cards under 4 children's chairs.

L5 3.1

Unless otherwise stated, all resources can be found on the main unit 3.1 page. From here, click on the icon to set a resource as a 2Do for your class. Use the links below to preview the resources; right-click on the link and 'open in new tab' so you do not lose this page.

- Lightning Scene.
- Moon Phases.
- Solar System.
- Viking Discovery.
- Unicorn Dog Seagull.
- Free Code Chimp (this is found on the main 2Code page).
- Storyboard Planner.
- Scene and Code Planner.
- Sketch Plan Example. Preparation
- Print and copy a range of planning documents for children to use in Lesson 5.
- Open the example programs: Lightning Scene, Moon Phases, Solar System and Viking Discovery in 4 browser tabs for easy access.
- Set Free Code Chimp for children to refer to in Lesson 5 and use in Lesson 6. You can select the following objectives when setting the 2Dos to make future assessment easier:

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

Year:	Y3	▼
Subject:	Computing	▼
Strand:	Computer Science	▼

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection and repetition in programs; work with variables and various forms of input and output.

Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

L6 3.1

None

L6 4.1

Resources Unless otherwise stated, all resources can be found on the main unit 4.1 page. From here, click on the icon to set a resource as a 2do for your class. Use the links below to preview the resources; right-click on the link and 'open in new tab' so you do not lose this page.

- Free Code Gibbon (this is found on the main 2Code page).
- Storyboard Planner. • Turtle Race game. Preparation
- Set Free Code Gibbon as a 2Do. You can select the following objectives when setting the 2Dos to make future assessment easier:

Year:	Y4	▼
Subject:	Computing	▼
Strand:	Computer Science	▼

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts.

Use sequence, selection and repetition in programs; work with variables and various forms of input and output.

Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

- Print copies of the Storyboard Planner for children to use.
- Create a display board for the class to share their programs to. Details of how to do this are given in Appendix 1

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

1. Unit 3.1 Lesson 3	<p>Deeper learning questions: How does coding relate to math? What skills do we need to debug?</p> <p>Reconnection: Go over online safety. Discuss prior learning. What can they remember about using purple mash.</p> <p>LO: To understand how to use the repeat command.</p> <p>Activity: Go over new vocab for lesson</p> <p>Display slide 4. Introduce the new vocabulary and discuss the piece of coding from the last lesson.(Not done so go over this.) Display slide 5. Explain to the children that they will be creating a program that uses the repeat command. Follow the instructions on the slide to start the scene. Use slide 6 to discuss and change the attributes of the button, including text, size, text colour and background colour. Display slide 7 and refer back to the aim of the task. Children to draw and think about the process of drawing a square. With slide 8, ask a child to walk in a square. Discuss the questions on the slide. Display slide 9. Talk through the slide and then open Free Code Chimp and drag across the repeat command to start your code. Note: If the children want to start with the button or when clicked, this is not incorrect, add that in and then add the repeat command inside the when clicked button event and continue. Display slide 10. Once the repeat command is set, we need to add inside it the block of commands that will repeat. Talk through the slide and then return to Free Code Chimp and drag in the object code block for your turtle object (mySuperTurtle1 if you haven't renamed it) and look at the possible actions for that object. Look at the available actions and ask children what they want the turtle to do 4 times. They are likely to say 'forward', and then 'turn'. That is fine to start with. Display slide 11. Talk through the side and then return to Free Code Chimp and add in the code similar to that shown and then test it (if children have asked you to add the button in already the code for the repeat command will look like this, but it will be nested in a when clicked button event). You'll see when you test this code, that the turtle moves in a square but doesn't draw anything. Ask children to help you fix the problem – by adding in turtle 'pen down' and turtle 'pen up' commands at the beginning and end of your block of commands. Test the code, watch the turtle draw a square, continue to test and debug if needed. Display slide 12. Return to the aim of the task – ask children: Does our algorithm achieve this aim? We need to program all this to happen when the button is clicked on (unless the children have already told you to do program the button). Return to your code and ask children to help you</p>
--	---

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

	<p>add the button in and move (drag) the repeat command into the when clicked event for the button. Test the code, click on the button, watch the turtle draw a square- or continue to test and debug if needed.</p> <p>Display slide 13. Talk through the slide and then demonstrate to children how you could develop the program by altering the image and the background. Remind children that they can alter the image of the turtle object by double-clicking on it, or by clicking on the image in the attributes table to open the clipart picker. Remind children how to rename objects.</p> <p>Introduce the activity with slide 14. Ask pupils to start 'Repeat Command' from their 2Dos and create a program that uses a turtle object and the repeat command. They can choose any background and change their turtle object into something else from the clipart gallery.</p> <p>Go over vocab and success criteria.</p> <p>Extension: Share the extension. Display an extension activity on slide 15. Ask children to see if they can develop their program by adding more objects and buttons, and explore what happens if they change the number of times the commands repeat or the angles the objects turn? Can they find out the number of sides and angles they'd need to set to make different shapes? – You could ask them to look it up or display them on the board for them to refer to.</p>
2. Unit 4.1 Lesson 4	<p>Deeper learning questions: When would if/else code be useful?</p> <p>Reconnection: Go over online safety. Discuss prior learning. What can they remember about using purple mash.</p> <p>LO: To understand the Repeat until command. To begin to understand selection in computer programming. To understand how an IF/ELSE statement works.</p> <p>Activity: Go over new vocab for lesson</p> <p>Start this lesson by returning to the Is It Raining 2Code example from lesson 2. Display the design view in one tab, and the Is It Raining IF Flowchart in another (or display slide 4). Ask the children to remind you what selection is and explain how an IF statement works.</p>

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

Explain that today we are going to start by looking at how to program something to happen if the condition is not met e.g. program something to happen in our scene if it is not raining.

Say to the children '**IF** my class is quiet for 30 seconds, **then** I will [insert action/ activity here!!] Start a timer and then check **IF statement** is true. If it is, carry out stated action/ activity. In pairs, ask one child to write an **IF statement** on their small whiteboard, then the other to check if it's true and run the action if it is, or not if it isn't. Discuss as a class: When tested, were any not true?

Explain that in code we can use **IF statements** to help our programs work – for example, **IF** the countdown has reached 0 the game is over, or **IF** the score equals 10 the 'amazing' sound plays.

- Load Reginald Rocket 2Code example. Look at the design first, and then the code. Look at the first part of the code and pick out the repeat until **command**.
- Can children 'read' the code to see what this **command** is doing?
When the user clicks on Reginald (the rocket), a message is printed to the screen – Prepare for Launch – then Reginald will move right (adding 1 to the X **attribute**) this repeats until the X is greater than the X position of Terry.
- Ask children what/who is Terry? (Terry is the launch pad) and what is the purpose of this first section of code? - This first section of code moves the rocket onto the launch pad.
- Now look at the next part of the code.
- What do children think will happen when this program is run?
- When the children click on Reginald he will move along to the launchpad. If the input is 'shoo' the sheep will run out of the way, if not (else) Reginald will take off with the sheep!

Show children the [Reginald If Else Flowchart](#) – if they look on the 'statement is true' side Reginald doesn't take off. What code would they need to add for Reginald to take off 3 seconds after the sheep was shooed away? Can we add a blast off sound?

Ask children to make a written plan with the following task specification:

Task: Create a short program that uses **Repeat Until** and **IF/ ELSE** commands.

Ask children to use the [Storyboard Planner](#) to plan their program. Challenge them to plan how their code will work using a **flowchart** on the back of their storyboard.

Remind children not to be too ambitious, and to think about the knowledge they have when making their plans, so they know they will be able to create them in 2Code.

Children could have [Free Code Gibbon](#) in front of them as they plan so they can look at available **backgrounds** and **objects**.

Children open the [Free Code Gibbon](#) task from their 2Dos area and start to make their plan into a working computer program.

Review children's work together against the lesson aims – this could be done by sharing some good examples from the 2Dos folder.

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

	<p>Go over vocab and success criteria.</p> <p>Extension: Share the extension.</p>
3. Unit 4.1 Lesson 5	<p>Deeper learning questions: When do we come across variables in real life?</p> <p>Reconnection: Go over online safety. Discuss prior learning. What can they remember about using purple mash.</p> <p>LO: To understand what a variable is in programming. To use a number variable.</p> <p>Activity: Go over new vocab for lesson</p> <p>Explain that today we will be working with variables. Go through the definition. Use this slide to help you explain what a variable is. On this slide each box is a variable. They both have names - the first variable is called 'team1score' and the second variable is called 'team2score'. The variable values are determined by how well (or not) each team does in a quiz. Put two actual boxes on a table so that all the children can see them, and label them 'team1score' and 'team2score' as in the slide. On the board, write 'team1score=' and 'team2score='. Read the IF/ ELSE statements that will have an impact on these variables – 'If a Team 1 answer is correct, the value of team1score will increase by 1, else the value of team1score will stay the same'. 'If a Team 2 answer is correct, the value of team2score will increase by 1, else the value of team2score will stay the same'. Tell children that they will be taking part in a class quiz that will have an impact on these variables. Split the class into Team 1 and Team 2. Team 1 play the first question. When they answer, check if it is correct and then react in the way the IF/ ELSE statement directs – use a number card to show the variable value in the team1score variable box (put the number card in the box) and add a value to 'team1score = ' on the board. Ask team 2 the next question – repeat as above until all the questions have been answered – changing the number cards in the boxes, and the values of the relevant variables each time an answer is given. Emphasise that the variable value is <i>replaced</i> with the new value each time – a variable holds only 1 value. Ask children to look under their chairs – four should find the Variable Game Cards you stuck underneath them. The cards say something like the following:</p>

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

	<p>Bonus points: add 4 to your team's score. Bonus points: double your team's score. Disaster card: subtract 2 from your team's score. Disaster card: halve your team's score. NB. At this point your variable value might increase to a number higher than you have number cards for, if this happens create the relevant extra values out of post-it notes or scrap paper! Who won? Discuss how the answers impacted the value of the variables and emphasise the importance of naming variables sensibly. Open the Genie activity. Stage 1: Complete together – when you create the variable point out that there are different types of variables and in this lesson, we are choosing 'number'. Creating the variable is a bit like making the box, our box in the classroom was named score, this one will be named 'count' as it keeps a count. When you click on Run to run the program point out the variable watch box. Explain that you can't see this variable in the scene as it's part of the code. Ask children to open Genie from their 2Dos and try and complete it independently. Remind them that they can click on the instruction to return to the video or unlock hints if they get stuck. Notes to support children with task. Stage 2: When children have added in the click event, they will need to add the 'change variable' code block and then select count because they need to change the count variable when the lamp is clicked on. When they click on Run to run the code encourage them to look at the variable watch box and notice how the variable changes each time they click on the lamp. See Y4 Coding Crash Course Lesson 6 plan for common mistakes children make at stage 3. Go over vocab and success criteria. Extension: Share the extension. Children have a go at working through Night and Day Gibbon that you have set as a 2Do.</p>
4. Unit 3.1	<p>Deeper learning questions: Why do we need to develop things?</p> <p>Reconnection: Go over online safety. Discuss prior learning. What can they remember about using purple mash.</p> <p>LO: To design and create an interactive scene.</p>

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

Lesson
5

Activity:

Go over new vocab for lesson

Display **slide 5** and introduce the main activity which will be spread over two lessons. Explain that the main aim of this lesson is to create a plan that would be good enough for someone else to follow to make your program if you didn't.

Display **slide 6**. Open the program, look at the design, the code and then run the program and discover how it works.

Give children time to discuss what they like or don't like about each program, and how they might **develop** it – if they think it needs **developing**.

Leave the program open in a tab.

Display **slide 7**. Open the program, look at the design and the code and then run the program and discover how it works.

Give children time to discuss what they like or don't like about each program, and how they might **develop** it – if they think it needs **developing**.

Leave the program open in a tab.

Display **slide 8**. Open the program, look at the design and the code and then run the program and discover how it works.

Give children time to discuss what they like or don't like about each program, and how they might **develop** it – if they think it needs **developing**.

Leave the program open in a tab.

Display **slide 9**. Open the program, look at the design and the code and then run the program and discover how it works.

Give children time to discuss what they like or don't like about each program, and how they might **develop** it – if they think it needs **developing**.

Leave the program open in a tab.

Display **slide 10** and talk through the object attributes for the animal object. Open up the program Unicorn Dog Seagull. Click on an object and look at the options in the **attributes** table with the children. How does changing each **attribute** affect the **object**? In the 'Lightning Scene' most of the **objects** in the design were hidden, and they were programmed to show when they were needed the **sequence**.

Set a sensible name for the objects in this scene. If 'allow off screen' is set to 'yes', the **object** will be allowed to move off the screen in the direction it's moving and it will disappear. If it is set to 'no' it will move off the screen in the direction it's moving, but come back on the other side, moving in the same direction. You could add a simple 'animal right' command to demonstrate this.

Display **slide 11**. Remind children that if there isn't an **object** type that matches what they want to add (e.g. they want to add a car) they can choose one that is there (e.g. animal) and then change the image and the name. If you return to 'Lightning Scene' and click on the fire engine you'll see in the attributes table that the fire engine is actually an **animal object** with the image and name changing it to a fire engine.

Display **slide 12** and ask children to read each example of code and **predict** what would happen when the program is **run**.

Go back to Unicorn Dog Seagull and program the second example: set the **show/ hide attribute** for the seagull to hide and then add code to make it show after 2 seconds and say 'Hi!' when it shows.

Emphasise the importance of **nesting** the **speak command** into the **after timer** or the seagull will speak but you won't be able to see it!

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

	<p>Display slide 13 and remind children of the button attributes. In Unicorn Dog Seagull, drag a button onto the scene and remind children how the attributes options are different.</p> <p>Display slide 14 and use it to remind children of the ‘Moon Phases’ program - notice how there are instructions at the start – explain that these were programmed using an alert.</p> <p>With slide 15, show the children the planning frameworks you’ve printed and copied and encourage them to choose a method of planning that suits them or that they think suits their program. They might favour a scene sketch like the one on slide 14, they could use the Storyboard Planner or the Scene and Code Planner they used in Year 2.</p> <p>Display slide 16. Give children time to make their plan, recommend that they have Free Code Chimp open in front of them. If they don’t finish this lesson they will be able to carry on in the next, and if they finish they could start making their programs in Free Code Chimp.</p> <p>Display slide 17. When their plan is complete, ask the children to open Free Code Chimp from their 2Dos and create their programs using their plans.</p> <p>(this plan and final design is lesson 6)</p> <p>Go over vocab and success criteria.</p> <p>Extension: Share the extension.</p>
5. Unit 3.1 Lesson 6	<p>Continue from above.</p> <p>Display slide 18. Remind children to save their work when they have completed it, and hand in their 2Do. Share children’s work to a Displayboard (see Appendix 1) and allow them some time to view and interact with each other’s interactive scenes. Review their work and celebrate achievements.</p>
6. Unit 4.1	<p>Deeper learning questions: Why is feedback helpful? What makes good feedback?</p> <p>Reconnection: Go over online safety. Discuss prior learning. What can they remember about using purple mash.</p> <p>LO: To review vocabulary and concepts learnt in Year 4 Coding. To create a playable game.</p>

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

Lesson
6

Activity:

Go over new vocab for lesson

- Display Turtle Game on the board, look at the design with the children.
- Click on 'rs' and 'ps' and look across to the **attributes** table – notice they are **number objects** with a **value** set to 0.
- Click on Run at the top to run the program and see that they display as 0 at the start.
- Click on the stop button and return to Design View. Click on the food, turtles and **button** and see how they are named in the **attributes** table. Ask the children to speculate as to how this game might be played.

Click on 'See Code' and look at the code, give the children a chance to 'read' it – you might want to split it into two parts as suggested in slide 6.

Give them time with a talking partner to discuss what the code will do.

Relate this to the teaching from the previous lesson on **variables** – what is going to change the values of the 2 number objects?

In the previous lesson the value of the variables was altered when the lamp was clicked on.

Rather than of **variables**, this program uses **number objects**, when the turtles collide with the food the value attribute of the object changes.

Click on Run to run the code and click on the red and purple race buttons in turn until one turtle has eaten all its food. Which turtle won?

The number objects keep a count of how much food the turtles have eaten, like a score.

Give children 2 minutes with their talking partner to discuss how this game could be improved. Ask them to feedback to the class and encourage them to consider if they think these are features, they could add in 2Code.

Use either children's feedback or this and the following slides to make some changes/ improvements to the game (or both!).

Open [Free Code Gibbon](#), click on 'Design' and add a **background** and **objects**, use the **attributes** table to name them appropriately. Add a **number variable** called 'score'. Last lesson we made created a **variable** in the code, this lesson we are going to add a **number variable** in design mode so the **variable value** (the score) is visible in our design.

Click on 'See Code' and add code that programs one **object** to collide with another and a score to increase. Add code so that the object collided into jumps to random **X/ Y co-ordinates**. Ask the children to **predict** what this will do.

Ask children to consider how this game might start and finish – how will the user know what to do?

Explain to the children that they will be designing and making their own game with a score.

Children use the [Storyboard Planner](#) to design their game and make notes on how it will work. They might like to be in front of [Free Code Gibbon](#) while they do this so they can explore the galleries and see what **backgrounds** and **objects** are available to them.

Once children have finished their designs, they have a go at creating them in [Free Code Gibbon](#). Remind them of the design – code – test – debug process.

These are some of the things they have learned so far – they might want to consider using some of them:

- **Number variables** to keep score

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

- **Selection** – IF and **IF/ELSE** statements
- Co-ordinates (using X and Y)
- **Timers** – after and every (could be handy to set a time limit)
- Repeat and Repeat until
- Alert boxes (these could be good for instructions)

Share children's work 2Displayboard (see Appendix 1) and play a few of the games together to share children's work and celebrate achievements.

Go over vocab and success criteria.

Extension: Share the extension.

End of unit quiz & reflect on gaps from the unit:

Unit 4.1 Quiz – found on unit page on PM (year 4)

Questions:

What is the term for adding a command that allows a program to choose a different outcome depending on a condition?

Which of these is a variable code block?

Tick all boxes where the statement matches the command.

Match the code block colour to the command type.

Which of these programs contains a conditional command?

Match the code block type to an example.

Place the object actions into the correct section of the venn diagram

Label the flowchart

Can you help CodeMonkey to debug this program to stop the rocket taking off too early?

Which of these actions can be assigned to a coordinate?

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

Match the words learnt in this unit with their definition.

End of unit quiz & reflect on gaps from the unit:

Unit 3.1 Quiz – found on unit page on PM (year 3)

Questions:

What is the term for placing one piece of code inside another piece of code?

Which of these is an output code block?

Place the object actions into the correct section of the venn diagram

Match the code block type to an example.

Which of these programs would make a bell ring twice every minute ?

Label the flowchart

Which of these are outputs?

Match the 2Code block types to their colour.

Which of these computer programs will draw a hexagon when a button is pressed?

Match the words learnt in this unit with their definition

End Points:

Why is it useful to use a flowchart to design a computer program? Using a flowchart to design a computer program is helpful as you can see it in its simplest form as inputs and outputs. You can see where the program is going which will prevent mistakes when creating the code.

What does repeat mean in computer programming? Using the repeat command will make a block of commands run for a set number of times or forever. This saves rewriting the code many times.

Pioneer Federation
Medium term plan
Cycle B, Term 6
ICT

What is the difference between 'timer after' and 'timer every'? A 'timer after' means after a certain amount of seconds, the action will occur. 'Timer every' means that the action will re-occur every certain amount of seconds on a loop.

Explain the stages of the design, code, test, debug coding process. This is a process to go through as you create a program using coding

- Design: create a design which could be a flowchart, a labelled diagram or a storyboard. This helps to think through the algorithms required
- Code: code the algorithms using to code and adapting the design.
- Test and Debug: see if the program works and fix any errors.

What does selection mean in coding and how can you achieve this in 2Code? The code will contain commands that require a decision and the next code to run will depend upon the outcome of this decision. In 2Code we used the 'if' command for selection.

How can variables and if/else statements be useful when coding programs with selection? The variable could be set either to 0 or 1 and this could be changed by user action or a timer. If/else statement outcomes could depend upon the value of the variable. command for selection.

What is the difference between the different object types in 2Code Gibbon level? The different objects have different properties. This makes them suitable for different type of programs.

- Buttons can only be clicked and have their colour and text changed.
- Vehicles have speed and angle.
- Characters have movement in 4 directions.
- Turtles have rotation, pen up and down.

Evaluation: What have the end of unit quizzes, pupil self-reflections and termly work told you about what the children can remember and recall? What are the gaps? Ensure that the areas that need further reinforcement are documented in the next subject unit MTP. **Plan in time to revisit gaps within units, determined by the quizzes.**

-