



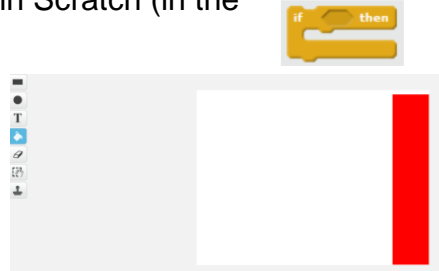



<p>Timing</p> <p>6 sessions of approximately 45 minutes.</p>	<p>Children will</p> <ul style="list-style-type: none"> • Make the Scratch cat move and change • Use Scratch to program keys and use them to 'draw' • Create a car racing game using their knowledge of 'if...then' commands in Scratch • Create their own game using their knowledge of Scratch
<p>e-safety links </p> <p>I can talk about why I need to ask a trusted adult before downloading files and games from the Internet</p>	<p>Objectives</p> <p>Programming</p> <ul style="list-style-type: none"> • I know that I need to keep testing my program while I am putting it together. • I can use a variety of tools to create a program. • I can recognise an error in a program and debug it. • I can use an efficient procedure to simplify a program. • I can use logical thinking to solve a problem by breaking it up into smaller parts. • I can use a sensor to detect a change which can select an action in a program. • I can recognise that an algorithm will help me sequence more complex programs.
<p>Links to other learning</p> <p>English: children can write a guide to their game, including instructions, for an audience.</p>	
<p>Resources</p> <p>Scratch 3.0 Desktop or online</p> <p>Bug video</p> <p>Selection video</p> <p>Etch a sketch game project</p> <p>Car racing game project</p>	<p>Preparation</p> <ul style="list-style-type: none"> • Use Scratch online or Download Scratch 3.0 from https://scratch.mit.edu/download. (If using Scratch online, make sure you have informed parents and talked through the responsible use of this online tool. A sample letter and details of responsible use can be downloaded from the website.) • Print out Etch a Sketch Predict handout for children to complete prediction task with programming blocks in session 2. • Make short-cuts to Etch a sketch game project , Cat, wall and fall project, Cat, wall and fall project and Car racing game project, or download to access on local network. • Have access to YouTube for Bug and Selection videos, and video showing use of an Etch a Sketch: <ul style="list-style-type: none"> ○ Video of child demonstrating how to draw a castle. ○ Video demonstrating line, circle and diagonal. • Print out RAG my design and algorithm poster • Print out blank planning template for creating a game in session 5. • Look at this document for more information on computational language e.g. selection and repetition in Scratch.

Expectations	Activity	Success Criteria
<p>Programming</p> <p>I know that I need to keep testing my program while I am putting it together.</p> <p>I can recognise an error in a program and debug it.</p> <p>I can use a variety of tools to create a program.</p>	<p>Make me move using the keyboard</p> <ul style="list-style-type: none"> • Talk in pairs about how you make things happen with Scratch software. What do you remember learning in year 3? Remind children of the importance of making mistakes and learning from them. Remind them how they learnt a lot from each other. • Explain that, in this session, we are going to control the cat sprite using the keyboard. • 3 block challenge: What can you make happen? You can ONLY use these three blocks. You can use them more than once. You can change the number and change the selection from the down arrow in point in direction block and when key pressed block.  <ul style="list-style-type: none"> • Once they have begun to explore make sure all the children have discovered the selection available from the drop-down arrows in the blocks. Stop the class and ask them what the choices make happen. Make sure they realise they can have more than one sequence in the script area at the same time. Encourage them to make different things happen when different keys are pressed. Can they create four controls to make the cat sprite move in four different directions? There are four directions available in the point in direction block. • You will need to intervene to let them add the 'set rotation style left-right' block from the blue motion blocks. (This will ensure that when the sprite is moved it stays the correct way up – or you can click on the i button on the sprite to change the properties there to side to side). You may want to suggest they also add 'if on edge bounce' block at the start of each sequence to stop the cat moving off the stage. • Explain to the children that we can also change the colour of the sprite. From the Looks blocks, drag the 'change colour effect by 25' block and connect it to the bottom of each movement stack. Now, when the sprite is moved around, it will change colour every time it moves. 	<p>Gold: Can I tell you my algorithm and create a program to control the cat sprite using the keyboard, produce a different sound when a direction key is pressed and say something when a direction key is pressed?</p> <p>Silver: Can I tell you my algorithm and create a program to control the cat sprite using the keyboard and produce a different sound when a direction key is pressed?</p> <p>Bronze: Can I create a program to control the cat sprite using the keyboard?</p>

	<ul style="list-style-type: none"> • Challenge: Can you use the keyboard to control the cat sprite and move it around the screen, changing colour at the same time? What is your algorithm? What is the sequence of things you want to happen? eg Press right key cat moves to the right. Press left key cat moves to the left. Press X key cat changes colour. • Remind the class of the term 'debug' to correct mistakes in a program. • Extension: explore having a different sound played for each key pressed and getting the sprite to say something for each direction chosen (e.g. "I'm moving left!" when the left key is pressed.) • Did your algorithm change as you began to program? Ask children to note down any changes they made to their original algorithm. 	
<p>2</p>	<p>Programming</p> <p>I can use a variety of tools to create a program.</p> <p>I can use an efficient procedure to simplify a program.</p> <p>Investigate an Etch a Sketch game</p> <p>There are many YouTube videos showing what you can do with an Etch a Sketch. Show one to the class and get them to talk about what is happening to create a picture. Video of child demonstrating how to draw a castle. Video demonstrating line, circle and diagonal.</p> <p>Predict</p> <ul style="list-style-type: none"> • Give all children Etch a Sketch predict handout. • Have a look at the algorithm. Look at the sequences of programming blocks. Ask children to read the sequences and predict what the programming blocks will make happen. Ask them to put their prediction in the blank boxes • Discuss the use of selection: if I press the up arrow key etc.  <p>Run</p> <ul style="list-style-type: none"> • Download this Etch a Sketch game or provide a link for children to use online. • Let children run the program. <i>If children are accessing a downloaded project, make sure they do a save as and give it a unique name before continuing.</i> 	<p>Gold: Can I predict what a sequence of programming blocks will make happen? Can I modify a program and explain the changes I have made?</p> <p>Silver: Can I run a program and investigate what each block does? Can I make changes to improve the program?</p> <p>Bronze: Can I run a program and tell you what I can make happen?</p>

	<ul style="list-style-type: none"> • Talk about the sequence of programming blocks for each key press as a procedure. <p>Investigate</p> <ul style="list-style-type: none"> • What happens if you change the numbers in the point in direction or move blocks? What happens if you change the colour? <p>Modify</p> <ul style="list-style-type: none"> • What can you add to improve your Etch a Sketch? Could be adding more colour choices. <i>To change colour choices click in the colour box in , and then click on the colour you want to select</i>  <p>Explain</p> <ul style="list-style-type: none"> • Ask children to explain the changes they made to each other. • Give children time to create pictures with their Etch a Sketch. 	
<p>Programming</p> <p>I can use logical thinking to solve a problem by breaking it up into smaller parts.</p> <p>3 I can use a sensor to detect a change which can select an action in a program.</p> <p>I can recognise an error in a program</p>	<p>Introduce the idea of ‘if something happens ... then ... (This is the idea of selection. For explanations for terms used when programming with Scratch see support sheet.)</p> <ul style="list-style-type: none"> • Prepare a child before the lesson to act out ‘if I walk into a wall’ then ‘I fall down’. <i>They must check after each step whether they are touching the wall. Draw the attention of the class to this as this helps the understanding that when they program this in Scratch they will need to use a forever loop to keep checking.</i> • Ask the class what is happening. Guide them to the use of if and then language. • Ask the children to find the if and then command in Scratch (in the control blocks). • Model painting a background which has a coloured wall. <i>(Select backdrop, draw a rectangle and make outline 0 to ensure it is just the fill colour as part of the backdrop)</i> • What will happen if the cat works across the stage and hits the wall? Talk about the 	<p>Gold: Can I describe how to use a sensor to detect a change?</p> <p>Silver: Can I use a sensing programming block to make something happen?</p> <p>Bronze: Can I make changes to a sequence?</p>

<p>and debug it.</p>	<p>algorithm you want to create eg</p> <ul style="list-style-type: none"> ○ Cat walks across the stage ○ If it touches the wall, then it will return to the start. <ul style="list-style-type: none"> ● Model how you would RAG this algorithm. Use this to draw attention to: <ul style="list-style-type: none"> ○ next costume will make it look like the cat is walking ○ sensing block for colour. (Click in the colour box and then on the colour on the stage you want to sense.)  ○ if then block must be within a forever loop. The sequence needs to keep checking if the cat is touching the colour, it can't just check once ● Tell the children you have had a go at implementing this as a program. Show them Cat, wall and fall project and Cat, wall and fall project using 'if then else'. ● Predict: What will each project make happen? What are similarities/differences? ● Run: Let children have a go at running the projects and ask to compare. ● Investigate and modify: Let children choose which project they want to use and make changes. Could add sound etc. ● Remind the children to keep trying things out and to debug any errors. ● Encourage children to look at each others' ideas and the programming they used. 	
<p>Programming</p> <p>I can use a sensor to detect a change which can select an action in a program.</p> <p>I recognise that an algorithm will help me to sequence more complex problems.</p>	<p>Make a racing car game</p> <ul style="list-style-type: none"> ● Ask the children to describe what selection is to a talking partner. Remind them of the work in the previous session and show Selection video if reinforcement of the concept is required. Refer children to selections (choices) they make in everyday life; for example, <i>if it rains in the morning, then I will wear my coat to school, otherwise I won't.</i> ● Create a chalk pathway on the playground. Ask a child to walk along and purposefully step off every now and then. When they step off they need to go back to the beginning. Rehearse the if and then language. ● Explain to the children that they are going to make a car racing game. ● Talk through the algorithm. <ul style="list-style-type: none"> ○ A car is driven along a road. ○ The car goes back to the beginning if it touches the grass. ○ When the car gets to the end of the course a sound is played and a message displayed 	<p>Gold: Can I read a program and predict what it will make happen? Can I identify programming blocks to use for a specific purpose?</p> <p>Silver: Can I read a sequence of programming blocks and predict what they will make happen? Can I modify a sequence to change the outcome?</p> <p>Bronze: Can I run a program and talk about what it makes happen?</p>

	<p>Predict</p> <ul style="list-style-type: none"> Look at the sequences in this Car Racing Game project. In pairs, ask the children to read each sequence and predict what it will make happen. Read each sequence together and take predictions for what it will make happen. <p>Run</p> <ul style="list-style-type: none"> Download the Car Racing Game project or provide a link for use online. Let children run the program. <i>If children are accessing a downloaded project, make sure they do a save as and give it a unique name before continuing.</i> <p>Investigate</p> <ul style="list-style-type: none"> Let children try out making changes to the programming blocks. Encourage them to talk to each other about the changes they make and what happens. <p>Modify</p> <ul style="list-style-type: none"> Tell the children they can make changes to the racing car game. Ask what the changes could be. This could be changing the background. Changing the amount of movement for each click of the arrows. They could use what they have learnt from the Etch a Sketch game to make it a two-player game. Allow time for the children to have a go at playing another child's game. Can they talk about the 'selection' options are that are being used? After playing the game, they could look to see if they are right. 	
<p>Programming</p> <p>I recognise that an algorithm will help me to sequence more complex problems.</p> <p>I can use a sensor to detect a change which</p> <p>5 & 6</p>	<p>Make a game (<i>Adapt the Car Racing game to create a game of my own</i>)</p> <ul style="list-style-type: none"> Provide a template for planning a game. You can download one of these: Box planning A4, Simple planning; or use a blank sheet of paper. Explain to children that today they are going to bring together what they have learned about selection to develop a game – they will have two sessions to plan, program and evaluate their game. Consider the planning process: <ul style="list-style-type: none"> What will happen on the screen? Plan the process / algorithm necessary to achieve this Carry out your plan Encourage children to think about how they could adapt the car racing game to a make a game of their own. Ask them for some similar scenarios e.g. Fox getting 	<p>Gold: Can I solve a problem by designing a program to achieve a specific goal?</p> <p>Silver: Can I use a sequence of steps to make things happen?</p> <p>Bronze: Can I select the bits of code to make things happen?</p>

<p>can select an action in a program.</p> <p>I can recognise an error in a program and debug it.</p>	<p>to the chicken house.</p> <ul style="list-style-type: none"> Ask the children to RAG their algorithm as a self-assessment to see what knowledge and skills they already have that they can apply <div style="border: 1px solid black; background-color: #e0f2e0; padding: 10px; margin: 10px 0;"> <p>RAG: Self-assessment / Formative assessment Support Poster</p> <ul style="list-style-type: none"> Ask children to RAG the algorithm, (highlighters or underline with coloured pencils). <ul style="list-style-type: none"> Red: 'I do not know which blocks to use to make this happen ... yet.' Yellow: 'I think I know what to do to make this happen but I am not sure. I am happy to have a go.' Green: 'I know which blocks to use to make this happen.' Use the self-assessment to identify children that may need: <ul style="list-style-type: none"> Red: additional direct teaching Yellow: Reinforcement Green: confident to work independently </div> <ul style="list-style-type: none"> Identify the children that may need further support / teaching and encourage those that are confident to begin programming independently. Remind them of the need to test as they program, to debug any problems. Stop the class when appropriate for different children to explain mistakes they have come across and how they have debugged them. Allow children sufficient time to complete their Scratch game. When complete, discuss outcomes with children as they play one another's games. What has been successful? What challenges did they overcome? What have they learnt? How did they learn it? Encourage children to think about ways they have collaborated and helped each other. What did they do when the program didn't work? Talk about the importance of making mistakes when you are programming. How many of them have learnt from making mistakes? 	
--	--	--