



Scratch - animated stories

<p>SAS Key Learning Objectives: I can design, write and debug programs and algorithms that respond to and learn from inputs I can design and create a program and debug it by collecting data and feedback from a group I can turn a programming task into an algorithm by identifying the important aspects of the task (abstraction) and then decomposing them in a logical way using my knowledge of possible coding structures. I can test and debug my program as I go and use logical methods to identify the cause of bugs, demonstrating a systematic approach to try to identify a particular line of code causing a problem.</p>	<p>Cultural Capital: Engaging with technology. Learning how technology works. Learning skills which will help in their future lives and careers and help participate in jobs of the future.. With a computer... they can create, manipulate, transmit, store and retrieve information and gain access to economic, political, educational, social, and cultural information. (Wakefield, 1986: 18)</p>	<p>British Values: Democracy- Evaluating, working in a group Rule of law- Evaluating, working in a group, following SMART rules Tolerance- Evaluating, working in a group, listening to others Individuality- Sharing ideas, creating work Mutual respect- Evaluating, working in a group, leading a group, respecting others online. Community- Collaborative work, Taking turns, Choices, Sharing, Listening skills</p>
<p>Cross Curricular Links: Maths - sequencing, formulas, expressions English - Story telling, writing dialogue</p>	<p>Reading: Reading texts onscreen</p>	<p>Resources: Scratch</p>



SAS Key Learning Objective (Outcomes)	Lesson Objective	Retrieval Task	Teaching Input and Activities	Challenge	Vocabulary
I can design, write and debug programs and algorithms that respond to and learn from inputs I can design and create a program and debug it by collecting data and feedback from a group	I can create appropriate animations for a story scene.	Recap Scratch and the areas of the platform	Show children the initial scene (characters and backdrop) in the Scratch Project: Haunted Castle 1a – Pupil Starter. Ask how the scene could be animated if it was part of a haunted castle story. What creatures or objects might be expected? Allow opportunity to discuss as a class, write down and feedback ideas (e.g. bat, ghost). Explain that children will be able to open this project and remix to add their own ideas	After the children have been given time to work, show ideas for the next steps to inspire development of their code. Sounds could be added, the size of a sprite can be changed and starting points defined for each sprite. Encourage children to add their own ideas and sprites.	Animate, animation, project, remix, repeat, iteration, debug.



<p>I can design, write and debug programs and algorithms that respond to and learn from inputs I can design and create a program and debug it by collecting data and feedback from a group</p>	<p>I can structure and control the timing of events.</p>	<p>Show the 'Haunted Castle 1b – Animated Example' project used in the previous lesson. Recap the sprites that have been introduced and animated. Explain that later there will be an opportunity to continue to code any sprites the children have not yet finished animating or to add more if they wish.</p>	<p>Establish that one of the next steps to improve the look of the animation is for the sprites to appear one at a time in sequence, rather than all at once. One way that we can control the order of events happening in Scratch is to use the 'broadcast' and 'receive' blocks. Show the effect of this with the next project example 'Haunted Castle 2 – Broadcast'.</p>	<p>Children use open-ended prompts to develop their code. Individual solutions to creating the each script may vary.</p>	<p>Animate, animation, project, remix, repeat, iteration, debug, broadcast, receive, sequence.</p>
<p>I can design, write and debug programs and algorithms that respond to and learn from inputs I can design and create a program and debug it by collecting data and feedback from a group</p>	<p>I can control when objects need to be visible</p>	<p>Children will have created and developed the code for multiple scenes in lessons 1 to 2 and should continue to build on these existing scripts.</p>	<p>There are two parts to this lesson. The first is to introduce the blocks 'show' and 'hide'. These can be inserted into scripts for each character sprite to make them only appear visible when</p>	<p>Children develop their code when adding a new backdrop and a transition button to switch backdrops.</p>	<p>Animate, animation, project, remix, repeat, iteration, debug, broadcast, receive, sequence, show, hide, visible,</p>



			<p>active. Demonstrate by running and then viewing the code in the project ‘</p> <p>The second part of the lesson introduces another use for the ‘show’ and ‘hide’ blocks.</p> <p>This is when you can create a button to switch backdrops. Demonstrate using the next stage of the project in</p>		<p>invisible, deconstruct.</p>
<p>I can design, write and debug programs and algorithms that respond to and learn from inputs</p> <p>I can design and create a program and debug it by collecting data and feedback from a group</p>	<p>I can sequence events to create a story narrative.</p>	<p>Children will have created and developed the code for multiple scenes in lessons 1 to 3 and should continue to build on these existing scripts.</p>	<p>Run the ‘Haunted House 4 – Story’ project for children to watch. Can children identify the new features or changes made? These include new backdrops/scenes, character speech and a story ending. Briefly look at some of the code added to make the story sequence work. What blocks are important to making the events work in sequence? What could be used to trigger the change</p>	<p>Children create their own script for an animated story, using their earlier ideas from the annotatedStory Planner Activity Sheet.</p>	<p>Animate, animation, project, remix, repeat, iteration, debug, broadcast, receive, sequence, show, hide, visible, invisible, deconstruct, transition.</p>



			of a backdrop? Recap how to use the arrow button and 'Broadcast' block, along with 'switch backdrop' block. Show examples from the 'Haunted House 4 - Story' script.		
I can design, write and debug programs and algorithms that respond to and learn from inputs I can design and create a program and debug it by collecting data and feedback from a group	I can add voice sounds to enhance an animated story.	Children will have created and developed the code for multiple scenes in lessons 1 to 4 and should continue to build on these existing scripts.	Explain that the next step in enhancing the animated story is giving voices to the characters. Audio files in Scratch are not just limited to the existing sound library – you can add your own speech, to make characters say whatever you like. Demonstrate with the 'Haunted House 5 - Audio' Scratch project.	Children use open-ended prompts and tips to create their own dialogue and recordings.	Animate, animation, project, remix, repeat, iteration, debug, broadcast, receive, sequence, show, hide, visible, invisible, deconstruct, transition, audio, record.



<p>I can design, write and debug programs and algorithms that respond to and learn from inputs I can design and create a program and debug it by collecting data and feedback from a group</p>	<p>I can add interactive user features to a scene or story.</p>	<p>Children will have created and developed the code for multiple scenes in lessons 1 to 5 and should continue to build on these existing scripts.</p>	<p>This lesson is designed to give some children a chance to finish off their previous scripts, while others can move on to creating an additional feature. If children have completed previous lesson tasks, they could add some interactive user features to their animated story. Interactive Features: What does this mean? The user is the person playing or reading the finished story. Interactive means the user is able to become involved by controlling some aspect of the animation. Show the 'Haunted House 6 - Complete' Scratch file and allow selected children to press one of the arrow keys to demonstrate the added features.</p>	<p>After the children have had the opportunity to add new sprites and associated code, it may be useful to stop the class and look at some of changes that would help the sequence of the story and allow time for the user to explore the interactive features. The arrow is used for transition to the next scene and an instruction box is shown in the story to explain what to do.</p>	<p>Animate, animation, project, remix, repeat, iteration, debug, broadcast, receive, sequence, show, hide, visible, invisible, deconstruct, transition, audio, record, interactive, user.</p>
--	---	--	---	---	---



Glen Hills Primary School Medium Term Planning

Computing

Year 6