Wandsworth CLC Computing

martin@wandsworthclc.com

Scheme Overview

In this scheme the students will be introduced to the Scratch Junior programming environment. They will learn to code the sprites used in the environment to move, play sounds, write speech and interact. They will understand the and use the words debugging and algorithm. Finally they will bring all these skills together to create a moving noisy text and sprite collage to demonstrate all the coding skills they have learned.

Lesson 1: Introduction to Scratch Junior

I Can Statement: I can program a robot or software to do a particular task.

Success Criteria:	**Key words:**
- Use code blocks to make a sprite move and turn.	Sprite, move, turn, code, block.

Overview:
In this step students will use scratch Junior code blocks to make a sprite move and turn around the screen. They will explore the basic functions of the Scratch Jr app.

Starter: Talk about whether computers are clever or stupid. Explain that computers only do exactly what we tell them, but they do many things very fast and this can appear to be very clever or even magic. Pretend to be a robot who can only understand move and a number and turn and a number. Challenge the students to try to move you around the room. Be VERY pedantic!

Teacher Input: 1
In this part the students will start to code a “sprite” to move around the “stage” by dragging code into the programming area. Point out the different areas on the screen and what each is used for. See Diagram on slideshow. To find out what a code block does you can tap on it before you use it and a description appears above the block.

Model dragging a ‘start’ block (yellow menu, green flag) and adding a ‘move’ block (blue menu, any arrow) onto the programming stage. Ask the students how far they think the sprite will move when you click the green flag (at the top of the screen) to start the program.

Show that one move command is quite small; can they suggest any methods for changing the program to make it a bigger move? You could:
add more to make the sprite move further
click on the block to change the number of times the sprite moves.

Model how to change the number of moves by tapping on the number box and typing a new number.
Show the students how to drag the sprite to anywhere on the screen by pressing, holding and dragging the sprite around the screen.

Student task: Timer 3 mins
Students to explore moving the cat around the screen; by editing the programming blocks and changing the starting position of the cat.

Teacher Input: 2
Next tell the students we want to be able to make the sprite 'turn' and 'move up and down'. See if any of the students can guess where the 'turn' code block will be (in the blue movement list.)

Model dragging a 'turn' block (blue menu, rotating arrow) to your existing program (connect it to the last command) and ask the students if they can suggest a way of increasing the amount of turns - you can change the number of turns by tapping on the number on the 'turn' command and changing it.

Test your program (press the green flag at the top of the screen).

Ask the students if they have noticed what happens when the sprite it gets to the edge; if necessary drag your sprite to the edge of the screen and press play - it will wrap around to the start of the screen, appearing on the other side.

Draw attention to the other blocks in the blue movement menu; 'move up', 'move down', 'move back', 'hop' and return home. Tell the students they will get some time to experiment with the movement commands.

Students will have an extra challenge; can you program the cat to move to all the sides and all the corners with one press of the green flag at the top left of the screen? If they manage that they can try again using hopping and get the sprite to return to the start to finish.

Student task: Timer 5 mins
Students to explore moving the cat sprite around the screen using the movement blocks.

Review:
Get students to describe the code they used and what it made the cat do.

Thinking Question:
What other things would you like the cat to be able to do?

Sample answer Speak grow shrink etc.

Lesson 2: Add a new sprite and change the background

I Can Statements: I can program a robot or software to do a particular task.

Success Criteria:	Key words:
- Add a new sprite and code it. - Choose and change the background.	stage, background

Overview: In this step the students will add new sprites (characters) to their code. They will learn how to change the background and how to run multiple code at the same time.

Starter: Timer 5 mins
Tell the students that we can use coding to create video games. Ask them if the video games they play have only one moving character.

Explain that to have a good game you need more than one moving character. Explain they are going to learn how to add more sprites (characters) and backgrounds to the code and have them working at the same time.

Teacher Input 1:
Show how to add a new sprite by clicking on the new sprite icon (the + sign) in the top left of the screen, choosing a sprite and tapping the tick.

Model how to change from one sprite to the other by clicking on each sprites icon in the top left of the screen.

Model clicking on the cat and creating a some code (e.g. 'green flag', 'move 10') to move the cat around the screen similar to the last step.

Can you think of a reason not to have the sprites start at the same time?
Stories happen in a sequence.
It would get to messy.

Show how the cat has code they have already written and the new sprite has none.

Model creating the code for the new sprite (e.g. 'green flag', 'move 10') and then show how both codes will run when you click the green flag and both sprites will move at the same time.

Can the students think of some characters they could use in an animated story

Student task: Timer 5 mins
Students to choose and add one new sprite and then create the code so that both the sprites move around the screen when the green flag is clicked.

Teacher Input 2:
Show the students how to add a new background by tapping on the new background icon (the picture icon) at the top, middle of the screen, choosing a different background by tapping on it and tapping the tick.

Student task: Timer 8 mins
Students to choose and add a new background to their code. See if the students can get both sprites to move to all sides and corners of the screen with one press of the green flag.

Review: Students to think of video games where they might need more than one character and different backgrounds.

Thinking Question:
How might you use more than one background in a game?

Lesson 3: Creating a racing game in Scratch Junior

I Can Statement: I can tell you the order (sequence) I need to do things to make something happen and talk about this as an algorithm .

Success Criteria:	Key words:
- Write code that animates multiple sprites. - Choose a suitable backgrounds.	Background, animate, move, algorithm

Overview: In this step the students will learn how to use multiple sprites to create a simple racing game and describe what they want to happen as an algorithm.

Starter: Talk about animation and how it is making something move and appear lifelike. See if the students can think of ways they could animate a character on the screen in scratch junior. Explain that the students need to create an animation of a car racing through a city street. Talk about how all the things they want to happen in the animation are a set of instructions that we call an algorithm.

Teacher Input 1:
Model creating an animal racing through a field. Describe this as an algorithm a set of instruction to make something happen. Add a suitable sprite and background. Model using the 'shrink' code block (it has an image of a person getting smaller on it) in the purple code blocks to change the size of the sprites. Create the code to make the animal race across the screen.

```
E.G. 'green flag'
     'Shrink 5'
     'Move forward 10'
```

Student task: Timer 5 mins
Students to create a car racing across the city. Get them to describe their algorithm.

Teacher Input 2:
Model using the 'set speed' code block in the orange code blocks (it has a person running and an arrow at the bottom) to change the speed of the animal. It can be walking, jogging and running.

Student task: Timer 8 mins
Students to explore changing the speed of their car and then to add 2 more cars with different speeds to recreate a race across the city.

Review: Ask the students what other animations they could think off that they could create.
Thinking Question: How might you use an animation of racing cars in the real world.
Sample answer: Starter screen for video games, adverts on T.V.

Lesson 4: Coding multiple sprites to create a moving landscape

I Can Statement: I can program a robot or software to do a particular task.

Success Criteria:	Key words:
- Combine growing shrinking and moving. - Code multiple sprites	control, grow, shrink, move

Overview: In this step the students will learn how to create a moving landscape of flowers that grow and shrink and move about the screen.

Starter:
Explain we are going to make our sprites grow, shrink, hide and move about the screen to make an interesting collage background.

Teacher Input 1:
Model adding a flower and using the 'grow and 'shrink' code from the purple code blocks to make them appear to grow and shrink. Introduce the 'hide' and 'show' code blocks also from the purple code blocks.

Show the students how to hide a sprite, code it to shrink and then use the 'show' and 'grow' to make it appear to grow again.

Can the students think of a way to slow the growing program?
Use a wait block.

Show them how to add a 'wait' code block from the orange blocks to make the growing more realistic.

EG 'Green Flag' 'hide' 'shrink 5' 'show' 'grow 5'

Student task: Timer 5 mins
Students to create code to make multiple flowers grow around the screen with a suitable background.

Teacher Input 2:
Model how to use a 'repeat' loop from the orange code blocks (it has a jigsaw shaped space in the middle, and a repeat number) to make the code easier to write.

Ask the students why we need to use repeat code blocks?
It tidies up the code making it easier to analyse and uses less programming power.

Drag the repeat loop into the coding area and then drag different code into the space in the loop.
Change the number of repeats by tapping on the number.
E.G. inside the 'repeat loop 5' 'wait 2' 'grow 1'

Student task: Timer 5 mins
Students to use the repeat loop to tidy up their flower code.

Review:
How do the students think they could use this sort of animation in a story?

Thinking Question:
How could they add a character to the scene and what could it do?

Sample answer: Move around pick the flowers smell them E.T.C.

Lesson 5: Adding a character to the scene and making it interact with the flowers.

I Can Statement: I can use the word debug when I correct mistakes while I program.

Success Criteria:
- Coding a character to interact with the flowers.

Key words:
Interact, when if, react. debugging.

Overview: In the step the students will add interaction between a character and the flower sprites.

Starter: Timer 5 mins
Get one of the student to pretend to be a small flower about to grow. Explain that you are a character in the forest and when you come across the flower and bump into it it will start to grow.

Model walking up to the student and bumping into them. Have the student start to grow. Explain the student is reacting to you, the character, by growing and this is called an interaction.

Teacher Input 1:
Model using the 'start on bump' code block in the yellow code blocks (It has two people facing each other and touching hands.

Create a simple flower grow program using a 'repeat' loop but instead of a 'green flag' start block use the 'start on bump' code block.

Ask the students if they any ways that we can interact with a computer program in real life?
Scanning food in a supermarket.
Typing pin codes in a cash machine. e.t.c.

Place a character on the screen in line with the flower and code it to move so it bumps into the flower (you can drag the character to the right place on the screen). Show the students how the flower doesn't do anything until the character bumps into it.

e.g. FLOWER 'start on bump' 'repeat 5' 'wait 2' 'grow 1'
CAT 'green flag' 'move 6'

(make sure you drag the cat to a place on the screen where he is horizontally in line with the flower and will move into it.)

Student task: Timer 5 mins
Students to explore creating flower code that starts on bumps with other characters.

When they have worked out getting one to work see if they can get 3 or 4 working.

Teacher Input 2:
Model adding 4 flowers that grow when bumped and model coding a character to move and bump into each one in turn making it grow.

Model how this takes quite a lot of trial and error to get the character to move to the right places around the screen and introduce the word debugging to describe this process.

Student task: Timer 8 mins
Challenge the students to create code for a sprite to move around the screen bumping into flowers and making them grow. Remind the students they will have to do a lot of debugging (trying out different code and changing it when it doesn't work) to get the sprite to move around to the different flowers and they will need to persevere!

Review: Share the different scenes created by the students and get the students to review each other's work with a 2 stars and a wish or some similar system.

Thinking Question:
Can you think of a story you might be able to use these animations to tell?

Sample Answer:
Little red riding hood springs to mind but the students may come up with many different ideas.

Lesson 6: Adding sounds and recorded speech to the scene.

I Can Statement: I can describe what actions I will need to do to make something happen and begin to use the word **algorithm**.

Success Criteria:
- Record a sound code block
- Get the sound block to play when the sprites interact

Key words:
sound , record, interact

Overview: In this step the students will learn how to record a sound block that can be used in the program.

Starter: Choose 5 different students and bring them up to the front. Give each of them a task such as moving in a small square, going pop, jumping up and down and moving in a zig zag.

Explain that the students will only start their tasks when you bump into them. Gently bump into each student to show how this works.

Tell the students that they are going create their own collage using different sprites that will move and make sounds as the characters move around interacting with them.

Teacher Input 1:
Show the students the 'play pop" sound block in the green code blocks (it has pop written on it.)

Show how we can code a sprite to use this by getting a sprite to move and then play the pop sound when it has finished.

E.G. 'green flag' 'move 5' 'play pop'

Ask if the students if they know why we would want sounds in our story or in any computer program?
The story can be recorded and then used by students who can't read yet.

Explain that we could have the sound play when the sprite bumps into another sprite

Model adding a 'play pop' code block to the previous step code. This will make the flower grow and play a pop sound when a character bumps into it.

Student task: Timer 5 mins
Students to add 'play pop' codes to their scenes.

Teacher Input 2:
Model recording your own sound blocks using the 'microphone' code block in the green sounds menu) by tapping on the block, tapping the record button, speaking into the

microphone (the microphone is at the opposite end of the tablet to the round menu button) and tapping stop when finished.

Model creating a recorded sound block and adding it to the previous move code in the same way you added a ‘play pop’ block.

Student task: Timer 5 mins
Students to use all the different coding they have learned to create a scene with growing flowers, sounds and characters interacting to start the code.

Review: Can the students recall all the different ways they can program the sprites in scratch junior and how the sprites will interact?

Thinking Question: How could we use the sound block in a story?

Sample Answer: Characters speaking to each other. Reaction sounds.