

## Stage, Sound & Movement

### Computing Science Concepts

- Unified start
- Sequence
- Some sequences have a definite order

### National Curriculum Programs of Study

Pupils should be taught to:

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**

#### Note

That this module hasn't got any specific links to the planning. You could build this in using the A3 planners or you could let pupils build and plan as they go.

set rotation style left-right ▼

## Stage, Sound & Movement Planning Version D

### [Teacher Demonstration]

#### Overview

1, Introduce the challenge of adding movement, sound and different backgrounds to an existing monologue or dialogue

2, Show pupils an example without going through the code

3, Demonstrate how to change character facings

4, Pupils add character facings

5, Demonstrate how to add background changes

6, Pupils create background changes

7, Demonstrate how to add sounds

8, Pupils add sounds

9, Demonstrate how to make characters hide or show

10, Pupils add hide and show if they want to

#### 11, Project assessment

1, Introduce the challenge of adding movement, sound and different backgrounds to an existing monologue or dialogue

Remind pupils of the monologues or dialogues that they created. Ask pupils what would improve the program? Backgrounds, sound, movement etc.

2, Show pupils an example without going through the code

Open up either the **bat example code (dialogue)** or the **Churchill example code (monologue)** in full screen mode. Can pupils spot what has been added?

3, Demonstrate how to change character facings

Open up one of their monologues or dialogues for all to see. Show pupils how to set the characters to face only left and right using the set rotation

style to left and right before dragging out a point in direction block and using 90 to point right and -90 to point left. Interspace this within the conversation .

#### 4. Pupils add character facings

Give pupils time to add their own character facings. Have they added them to add to the dramatic or to link in with what is being said?

#### 5. Demonstrate how to add background changes

Show pupils how to import backgrounds using the choose a backdrop tool. You may also want to show them how they can draw a backdrop and import a backdrop using the buttons above.

Give pupils time to import backdrops that go with their dialogue or monologue.

Then show pupils how they can change backdrops using the switch to backdrop block in either the sprite or the stage area. If using the stage area you will need to add time blocks.

#### 6. Pupils create background changes

Give pupils time to add their backdrop changes into their programs. Do their changes fit with the text?

#### 7. Demonstrate how to add sounds

Navigate to the sound tab. Show pupils how to add sounds and then how to start these using the start sounds blocks. Demonstrate placing these within the dialogue or monologue. It can be helpful to remind pupils that you only import a sound into a specific area so it will not be in other sprites unless they import it manually.

#### 8. Pupils add sounds

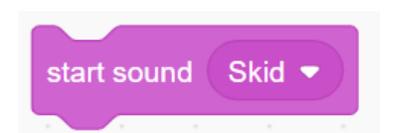
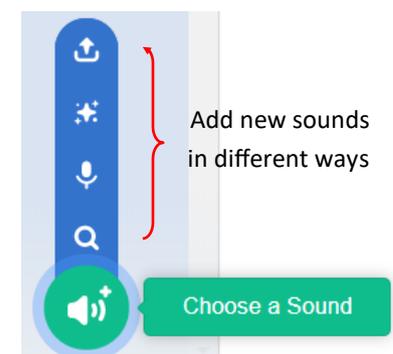
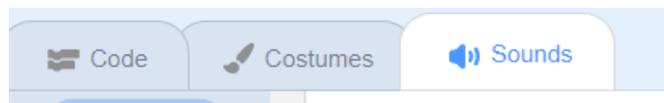
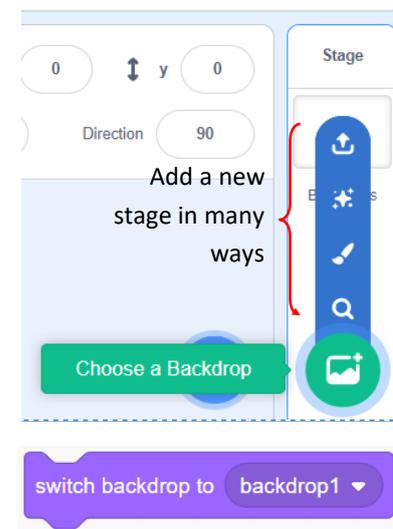
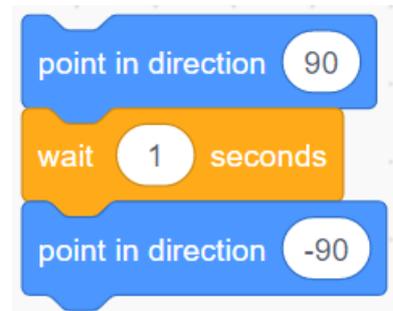
Give pupils time to add sounds. Do their sounds go with their themes?

#### 9. Demonstrate how to make characters hide or show

Show pupils the hide and show blocks. They can figure the rest out themselves. Remind them to make sure the character is shown when they start the program as pupils can easily hide and forget to add a show block at the start.

#### 10. Pupils add hide and show if they want to

Give pupils time to add this feature if they wish





Two stars and  
a wish

## 11, Project assessment

### Possible Summative assessment opportunities

You could assess the final finished project.

You could assess the planning.

You could assess how far the planning has been fulfilled in the final project.

You could ask pupils to tell you what a good version looks like (WAGOLL), collect their ideas and then get them to assess their own projects against the criteria.

Pupils could also assess their computational attitudes using some of the boxed statements over the page.

You can find a summative assessment quick **Kahoot Quiz** linked at <http://code-it.co.uk/gold/>

Whilst Kahoot is a limited assessment tool it is free and it is easy for teachers to pass the results back to code-it via [phil.bagge@code-it.me](mailto:phil.bagge@code-it.me) so I can look at which method provides best short test results. Not conclusive but useful.

If you pass the results back please

- 1, Anonymise the results by removing the names
- 2, Ask the head teacher for permission
- 3, In the email title say which module you are doing (ie Animation D)

*I recognise there is more than one way to solve/describe a problem*

*I can evaluate my solutions against a set criteria*

*I can design criteria to evaluate my creations*

*I can contribute useful ideas to a partner or group*

*I can encourage others to share their ideas*

*I lead using all the people talent in my group*

*I learn from setbacks and don't let them put me off*

*I can persevere even if the solution is not obvious*

*I don't just accept the first solution*

*I look for a range of solution to the same problem*

*I look for how a project can be extended*

*I can break complex problems into parts*

*I can discover / concentrate on the most important part of a problem*

*I can identify patterns in problems & solutions*

*I can adapt existing ideas to solve new problems*

*I can develop, test and debug until a product is refined*

*I make predictions about what will happen*

*I repeatedly experiment through predicting, making, testing & debugging*



Handles Ambiguity



Open Ended Problem Solver



Evaluates

Insert picture of your students here



Copes with Complexity



Communicates



Adapts



Perseveres



Investigates