Curriculum objectives
- To design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. To use sequence and repetition in programs; work with variables and various forms of input and output. To use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Lesson objectives
- To use multiple variables to add scores, timers, lives and so on as appropriate.
- To test and debug their code as they write it.
- To work collaboratively to debug their code.

Expected outcomes
- Can use multiple variables where appropriate to create an effective game.
- Can work collaboratively to test and debug their programs.

Resources
Completed planning from previous lessons; photocopiable page 181 'Variables and procedures'

Adding scoring, timers and lives

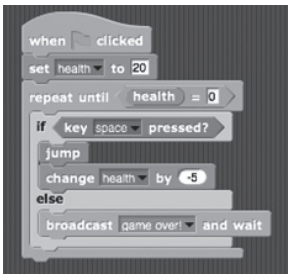
This week, the children are reminded of the use of using multiple variables, such as scores, lives, and timers in their game. They are also introduced to using a variable for health. They combine the use of variables with their knowledge and understanding of procedures, to create procedures for their variables as appropriate in their game, and continue to work on finishing their game, collaboratively testing and debugging their code.

Introduction

- Remind the children that in the last lesson they were continuing to program their game and created their own procedures. Lead a brief discussion to recap how and why procedures are used and how to make blocks in Snap!
- Explain that this week they will be working with their game variables – you will be showing them how to create a variable for health, if appropriate to their game, and they will be creating procedures for their own variables.

Whole-class work

- As a class, work through how to create a health variable. By now, the children should be able to work this out with a little guidance. First, make a variable called 'health'. The screenshots show how health can be set and reduced every time the player jumps.



- As a class, talk through the code and how it is working (every time the space key is pressed, the procedure 'jump' is called and health is changed by −5. When health reaches 0, 'game over' is broadcast. Ask: *How many jumps can be used in the game? Is this good gameplay?*
- Give out photocopiable page 'Game variables and procedures' and go through the example with the children.
- Explain that you are now going to show them how a procedure can be used to increase health, for example when the character comes into contact with different energy-boosting items. This is a good time to use a procedure as we can call on it whenever we want an object (sprite) to be able to give an energy-boost to the character. The screenshots below show a possible script for the procedure and different ways that it could be cued.

Paired work

- Ask the children to plan their own variables on photocopiable page 181 'Variables and procedures' and identify where it would be useful to create procedures.
- For the remainder of the lesson, the children should program their variables into their game, as usual working collaboratively to debug any errors.

Differentiation
- Support: Less confident learners may need additional support in understanding how variables work and planning and programming their own variables. You could encourage them to stick to one variable if necessary and you may feel that they do not need to create any further procedures if they found this challenging last lesson.
- Challenge: More confident learners can plan and use multiple variables and procedures with confidence. Encourage them to help others debug their work.

Review

- Ask volunteers to share how they are using variables in their work and highlight where they have created procedures that include their variables.