

Year: 6
Term: Spring 2
Topic: Programming - Variables in games



Programming is when we make and input a set of instructions for computers to follow.

We use **algorithms** which we can plan, model, trial and debug, in order to create accurate command sequences, that enable variables to be enacted in games.

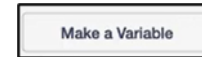
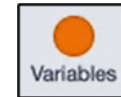
Variables: A variable is something that is changeable. A variable can be set and changed throughout the running of a program.

In computer programming we use variables to store information that might change and can be used later in our program. E.g. in a game a variable could be the current score of the player; we would add 1 to the variable whenever the player gained a point.



Making Variables in Scratch – The Basics

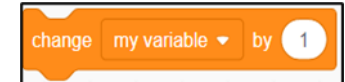
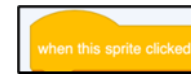
Select 'Variables' (dark orange circle). Either choose from the available variables or 'Make A Variable'.



Select 'Events' (light orange circle). Choose what needs to happen for the variable to change. E.g. 'When this sprite clicked' or 'when space key pressed.'

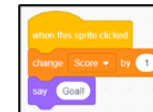
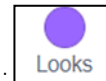


Select 'Variables' again. Choose what will happen when the event happens, e.g. 'change score by 1' (to add a point) or 'change score by -1' to remove a point.

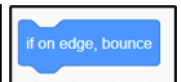


Variables should always have a value and an appropriate name.

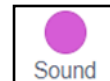
Adding Callouts: Select 'Looks' from the menu. Add it to the variable program. Edit the text to change the callout.



Adding Motion: Many games require sprites to change position. This is achieved using the 'Motion' commands. Select 'Motion'. Choose from the available motion commands.



Adding Sound: Many games require sprites to change position. This is achieved using the 'Sound' commands. Select 'Sound'. Choose from the available motion commands



Adding Comments: Comments are a good way of showing that you understand what your code is doing. Right click on the block that you want to comment on, and add in your comment.



Programmers do not put their computer programs straight to work. They **trial** them first to find any errors:

Sequence errors: An instruction in the sequence is wrong or in the wrong place.

Keying errors: Typing in the wrong code.

Logical errors: Mistakes in plan/thinking.

If your algorithm does not work correctly the first time, remember to debug it!

