

# Does writing algorithms improve pupils' chances of porting their knowledge?

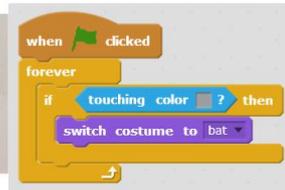
## Brief Summary

If you like maths  
stand  
else  
wave

Everyday algorithm  
roleplay & writing

forever is touching  
grey change sprite  
to bat

Writing algorithms before and during coding



Forever  
's button pressed  
Red light  
Show  
Else  
Turn green

Increased ability to use algorithmic  
concept to solve other problems

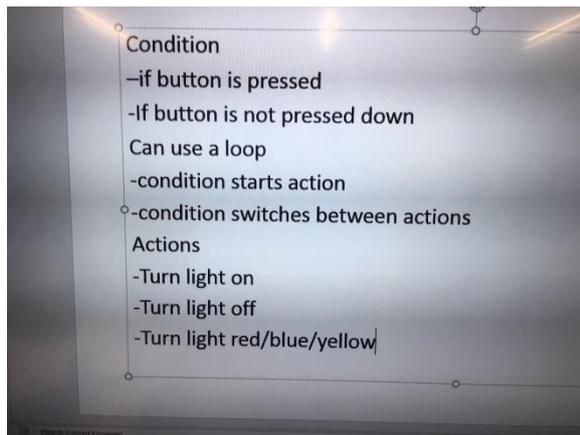
## Outline

- 32 Year 5 pupils
- Pupils spent 30 minutes learning about selection roleplaying and writing everyday algorithms.
- Pupils spent 20 minutes on roleplaying and writing shape selection algorithms where the emphasis was on selection within a loop.
- When planning their own Scratch game pupils wrote short selection algorithms before coding.
  - This was a process most pupils repeated three times and some many more.
- Pupils spent the remainder of the six hours programming their own game.
- Approximately 30 minutes was spent in use, modify, create code examples half way through the project.
- The teacher used a mixture of coding and non-coding language when modelling algorithms as he was transitioning to using non-coding language in the planning/algorithm level.

## Assessment

Pupils were asked to write an algorithm to make a button do something on the Crumble microcontrollers

- Pupils were shown a crumble microcontroller attached to a Crumble playground with a button and light plugged in that were non-operational.
- 6 pupils had used the Crumbles before in an earlier Year group, but none had used it with selection.
- They were given these helpful hints on the board which were read out but not expanded upon apart from the last point where it was explained that the light can display lots of different colours using the same bulb.



## Findings

- 16 pupils wrote a condition chooses action that was recognisable
- 13 pupils wrote a condition switches between actions that was recognisable
- 29 pupils wrote a condition that was recognisable
- 14 pupils also added a loop that was in the right place. Most wrote this as a forever loop and some drew a looped arrow.
- 1 pupil included a loop inside the condition
- 13 pupils didn't include a loop
- 3 pupils used non-Scratch language like loop
- 2 pupils wrote conditions that were nothing to do with the task, these were everyday condition starts action
- 2 pupils didn't include any selection and mostly copied the helpful hints
- 1 pupil wrote nothing
- When they turned their algorithms into programming only one pupil was reminded about the necessity of their programming needing a loop by the teacher.

## Example algorithms

condition  
 is up arrow go up  
 is done arrow go up  
 actions  
 - Turn light on  
 - Turn light off

```

  if button pressed
  Forever
  turn on light
  if key not pressed
  Forever
  light don't go on
  
```

Forever  
 If button is not pressed  
 Turn light orange  
 else  
 Turn light green

If <sup>Forever</sup> button is pressed?  
 turn light on blue  
 else  
 turn light on red

IF Button <sup>not</sup> pressed change color if not  
 stop  
 if Button <sup>not</sup> pressed turn light on

Challenge  
 Forever  
 if button pressed  
 Red light  
 Show  
 Else  
 Turn green

Forever  
 press button turn light on  
 or  
 don't press button turn light off

Forever  
 if ~~space~~ button pressed  
 light on  
 wait 2 sec  
 light off  
 wait 2 sec  
 turn light red  
 wait 1 second  
 turn light blue  
 wait 1 sec  
 turn light green

if button pressed  
 Turn light blue  
 Else  
 Turn light green

if button is pressed  
 do floss and  
 if you pressed  
 down 2x and if do you  
 sit

If button pressed  
 Turn light blue  
 If button not pressed  
 Turn light green

IF button pressed — condition  
 then — and go  
 turn on blue — then/do  
 else — else/or  
 stay off

forever  
 if button pressed down  
 turn light on  
 else  
 if ~~light~~ ~~is~~ ~~switch~~ turn light off  
 forever  
 if button pressed twice within 5 seconds  
 Turn light red

Teacher conclusions

- The number of pupils who understood what selection is and how it works was higher than for previous classes. 27 pupils wrote meaningful selection algorithms to solve the problem even if 13 of them missed out the loop.
- Time spent thinking about selection in the planning/algorithm level translated into a better ability to port this idea between programming languages which has been an issue with other classes.
- Scribe for those that are poor writers or provide cards they can manoeuvre/write on

Limitations

- Very small study
- No control groups
- Only selection and selection within a loop

#### Next Steps

- Test the same thing with other classes
- Test with other algorithmic concepts such as repetition or variable use

Phil Bagge

December 2018

