PROGRESS OUTCOME 5

# Mixing colours

## Annotation

Pita is able to apply his knowledge about the digital representation of colours to develop a computer program that includes a calculation for a 24-bit colour with red, green, and blue values. He can use compound mathematical operations in a computer program and variables to represent and store data, including the results of calculations.

His program uses:

- sequence (steps in the correct order)
- inputs (RGB colour values, pen size)
- outputs (colour, pen-size, and text displays)
- selection through conditional logic (if–then-else)
- iteration (a forever loop).

He has also supported his program with comments that explain his code and calculations.

## Background

The students in Pita's group have been learning programming in a block-based programming environment. They have developed, tested, and debugged several programs that use the programming constructs required for the task below (sequence, selection, iteration, inputs/outputs, conditional logic, and compound mathematical calculations). They have also developed a basic paint program that uses set colour choices.

In addition, the students have been learning about binary representation for colours using 24 bits (8 bits for red, 8 bits for green, and 8 bits for blue) and how colours are displayed on a screen using pixels. They have applied the calculation used by computers to determine colours.

## Task

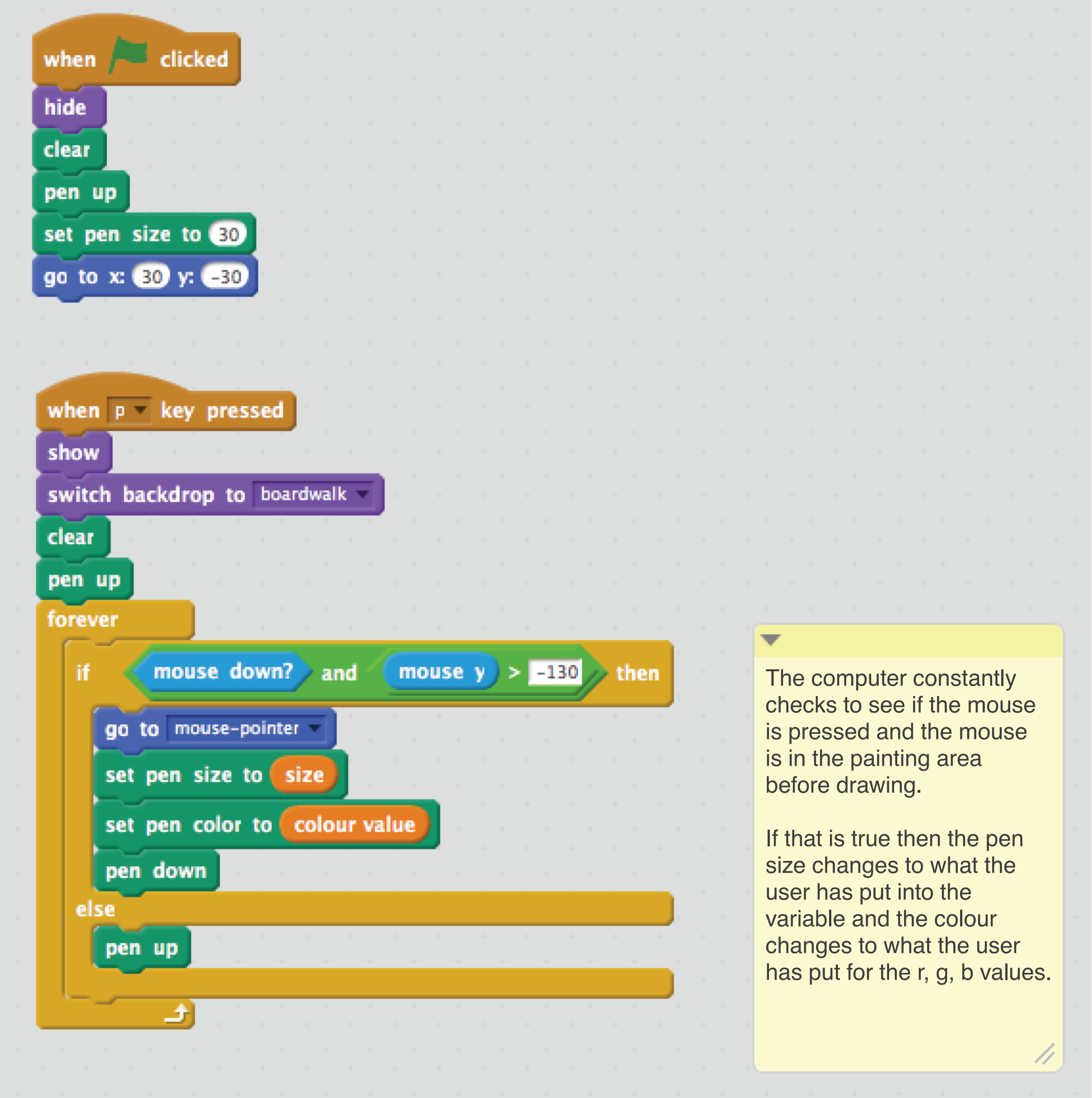
The students are asked to refine their basic paint program so that the user can create custom colours to paint with on the screen. The user should be able to input the RGB (red, green, and blue) values for a colour, and the program should then calculate the colour.

The program should include variables for red, green, and blue values (between 0 and 255), the colour value, and the size of the paint brush. It should also display a "swatch" of the custom colour that the user has mixed, to allow them to preview their colour.

**There are clear links to mathematics in the calculations that underpin colour values. The task is of particular interest to students studying photography and developing their ability to print a digital copy of an image that accurately captures the colours of the original.**

## Student response

Pita develops, tests, and debugs his program and adds relevant code comments. In the following code, he uses inputs from a user (pressing the P key; mouse clicks), sequence, and selection (if-then-else), and he demonstrates his understanding of the x and y coordinates of the mouse. He also uses conditional logic to determine whether the mouse is within the drawing area of the program's canvas.



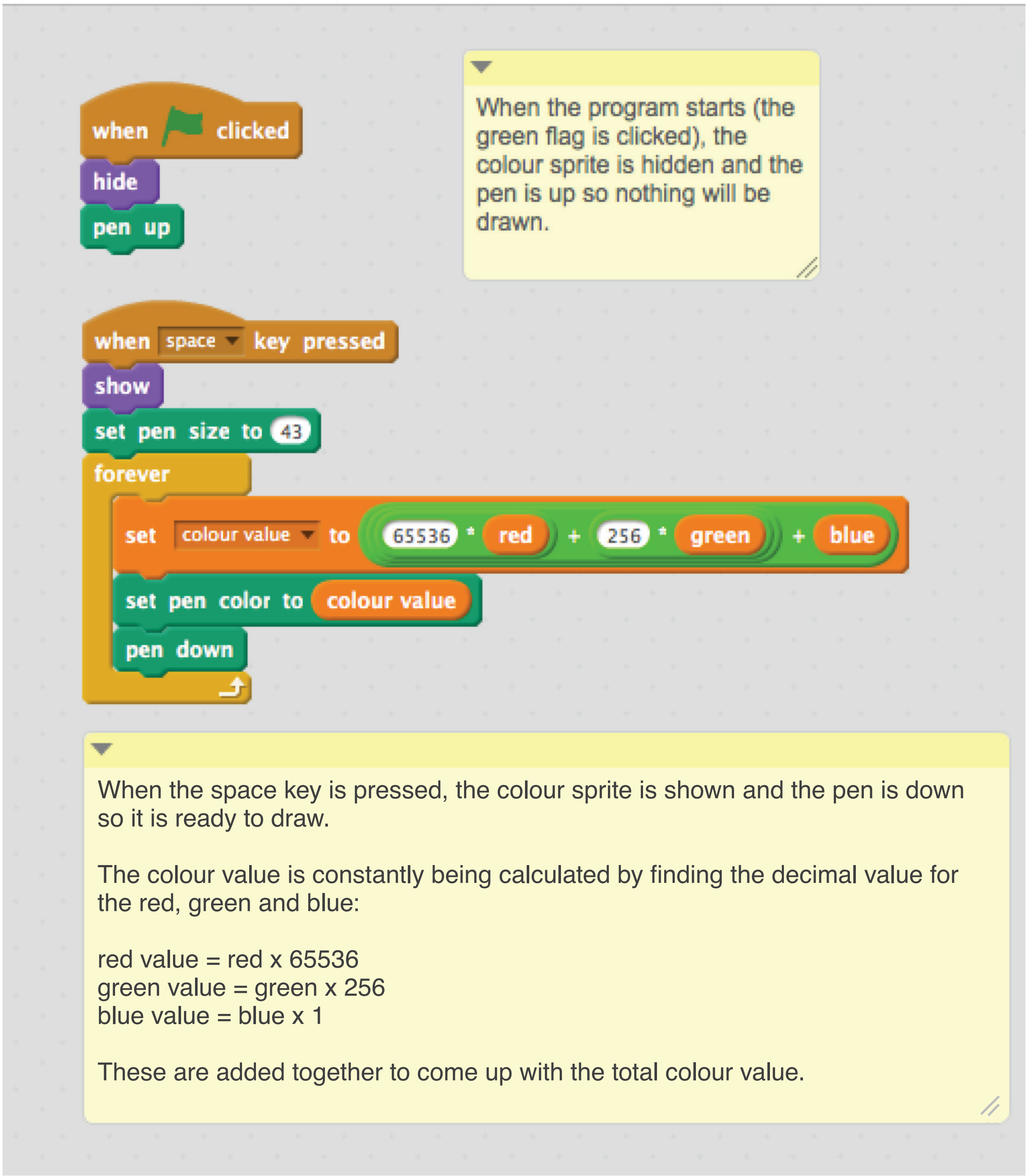
```
when ⚑ clicked
hide
clear
pen up
set pen size to (30)
go to x: (30) y: (-30)

when [p ▾] key pressed
show
switch backdrop to [boardwalk ▾]
clear
pen up
forever
    if <<mouse down?> and <(mouse y) > [-130]>> then
        go to [mouse-pointer ▾]
        set pen size to (size)
        set pen color to (colour value)
        pen down
    else
        pen up
```

The computer constantly checks to see if the mouse is pressed and the mouse is in the painting area before drawing.

If that is true then the pen size changes to what the user has put into the variable and the colour changes to what the user has put for the r, g, b values.

In Pita's colour-swatch code below, he uses variables (to store data and the results of calculations) and iteration (a forever loop). He also applies compound mathematical operations within the program and includes code comments to document how his code works.



```
when green flag clicked
hide
pen up
```

When the program starts (the green flag is clicked), the colour sprite is hidden and the pen is up so nothing will be drawn.

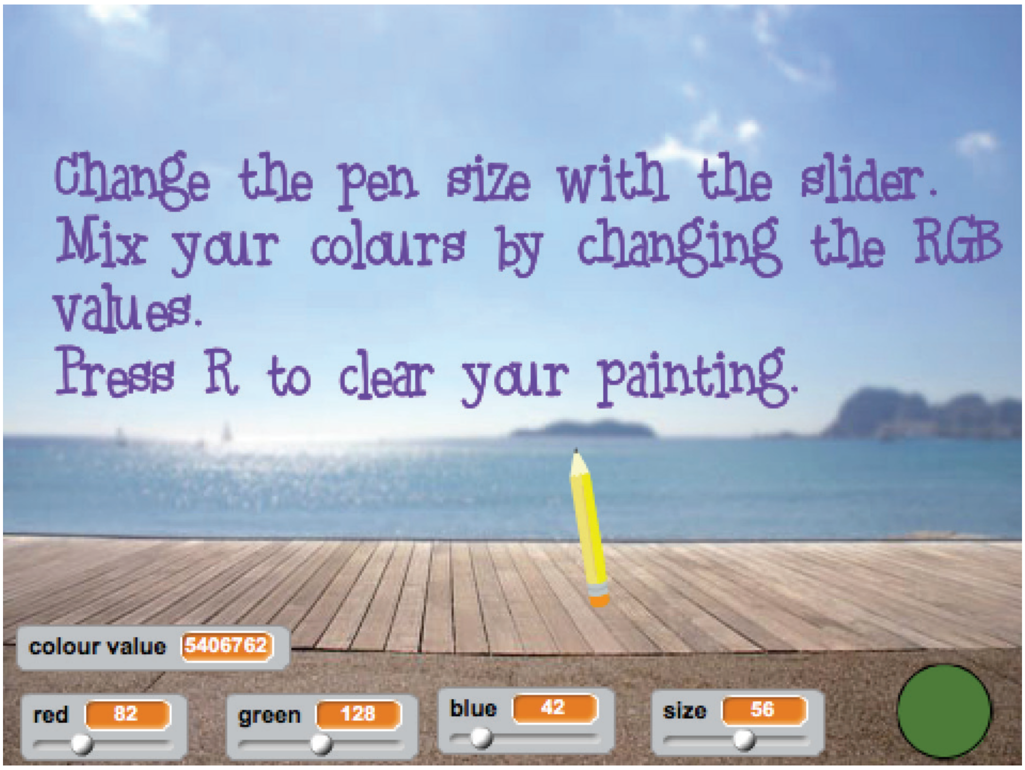
```
when [space] key pressed
show
set pen size to (43)
forever
    set [colour value] to (((65536 * (red)) + (256 * (green))) + (blue))
    set pen color to (colour value)
    pen down
```

When the space key is pressed, the colour sprite is shown and the pen is down so it is ready to draw.

The colour value is constantly being calculated by finding the decimal value for the red, green and blue:

red value = red x 65536  
green value = green x 256  
blue value = blue x 1

These are added together to come up with the total colour value.



Change the pen size with the slider.
Mix your colours by changing the RGB values.
Press R to clear your painting.

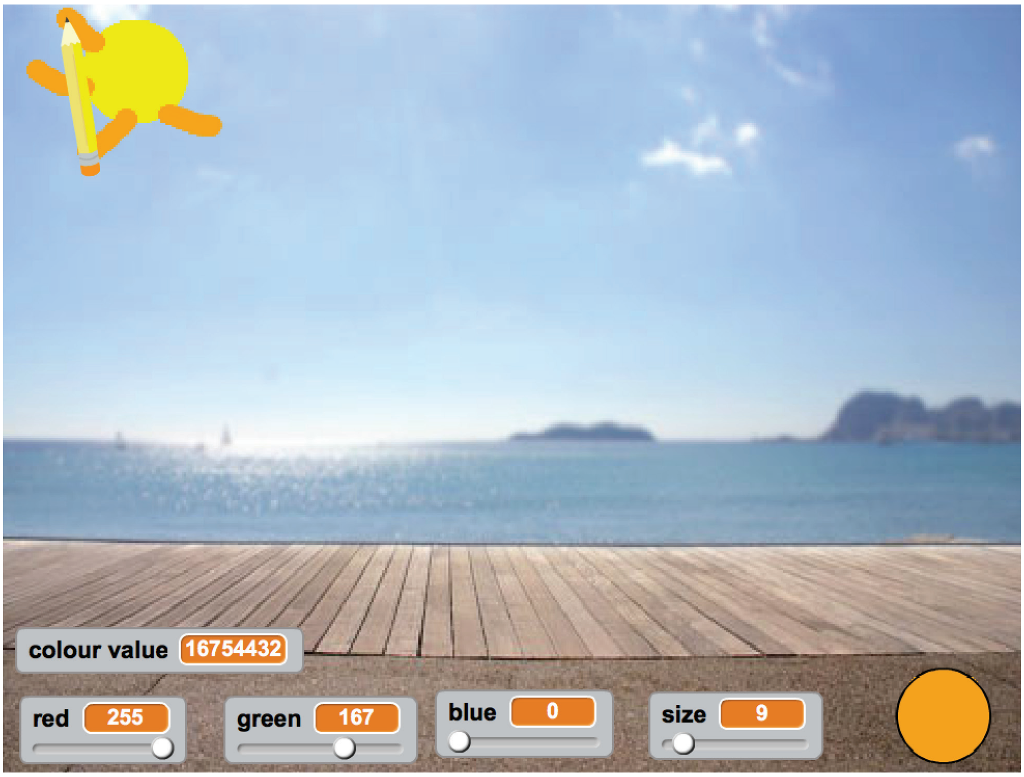
colour value 5406762

red 82

green 128

blue 42

size 56



colour value 16754432

red 255

green 167

blue 0

size 9

Downloaded from http://technology.tki.org.nz
Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab (http://scratch.mit.edu).

The program used in this exemplar is not officially endorsed by the Ministry of Education.

ISBN: 978-1-77669-209-5

MINISTRY OF EDUCATION
TE TĀHUHU O TE MĀTAURANGA