Microbe battle

Scratch lesson plan – Code Playground

BARCLAYS

Lesson overview

In this lesson, students will create a Scratch project called Microbe battle, where two microbes compete to determine a winner. This project introduces students to efficient coding by limiting the number of lines per sprite. The aim is to help students think creatively about achieving functionality with minimal code. Teachers will guide students through creating sprites, adding scripts, and testing their game, encouraging problem-solving and reflection.

Time	Key learning outcomes	Resources
45 mins	• Understand how to create and customise sprites in Scratch • Develop scripts using a limited number of lines of code • Explore efficient coding by optimising block usage • Apply creativity and problem-solving under constraints.	• Laptops or desktop computers • Access to Scratch website - https://scratch.mit.edu.

Content

Activities	Time	Page
Activity introduction	15 mins	3
Activity – Microbe battle	30 mins	4
Code snippets	-	5
Summary	-	6

Introduction

The microbe battle project was created to highlight the importance of efficient coding. When programming there are often multiple ways of creating the same effect. Some of these will require large amounts of code, by optimising how a project is coded and finding the shortest way of writing the code will save time but also reduce complexity hopefully avoiding any bugs.

"In coding, doing more with less makes your code easier to understand and faster to run"

The activity of this programme is to copy the code onto created sprites as a way to demonstrate how efficient coding can be. Using less than 20 lines of code to run the project. More advanced coders should be encouraged to apply this knowledge to previous coding projects. What ways can they find to optimise their existing code projects.

Hold a class discussion on the topic of making instructions as brief and as simple to follow as possible. What do students think could be the impact of longer and more complex instructions on how a computer runs and might handle these instructions?

Scratch practical

Ask the children to log into Scratch and set up their workspace as described in the workbook. Show the microbe battle video as a guided lesson pausing regularly when the pupils need to catch up.

- Children should be able to follow along with the workbook or the guided lesson video
- By the end of the lesson children should be able to create a working program showcasing the Microbe battle activity.

Activity – Microbe battle

“Today, we’ll create a Scratch project called Microbe Battle. In this project, you’ll program two microbes to compete, but you’ll only use 20 lines of code per sprite”

Class question:

“How can small changes improve the player’s experience while keeping the code efficient?”

Encourage the pupils to customise their own backdrops and sprites to make their project unique.

Scratch practical

Using the video and workbooks support the children to follow the instructions and complete the coding project. Have them think of other ways to enhance the project if they have extra time.

Activity wrap up

Prepare to share your project with the class.

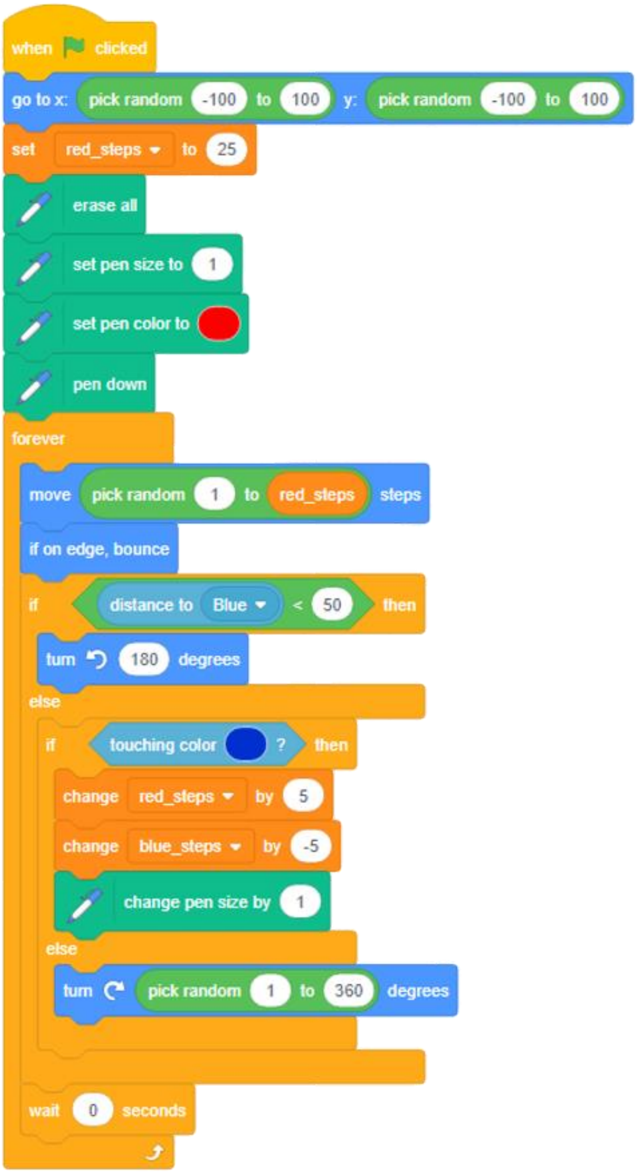
“What was challenging about limiting the number of lines in your script?”

Encourage customisation

- Add a timer or scoring system using 5 additional lines of code
- Optimise a previous project by reducing redundant blocks
- Experiment with adding more microbes while maintaining the limit for each sprite.

Code snippets

Sprite code



```
when 🏁 clicked
go to x: (pick random -100 to 100) y: (pick random -100 to 100)
set red_steps ▾ to 25
erase all
set pen size to 1
set pen color to ●
pen down
forever
    move (pick random 1 to (red_steps)) steps
    if on edge, bounce
    if <(distance to Blue ▾) < 50> then
        turn ↺ 180 degrees
    else
        if <touching color ● ?> then
            change red_steps ▾ by 5
            change blue_steps ▾ by -5
            change pen size by 1
        else
            turn ↻ (pick random 1 to 360) degrees
    wait 0 seconds
```

Summary

The following information is an example of what a child at an expected level would be able to demonstrate when completing these activities with additional examples to demonstrate how this would vary for a child with emerging or exceeding achievements.

Assessment guidance

Differentiation – Lower ability/ASN	Differentiation – Higher ability/extension
• Provide pre-made sprites for students who need extra support • Simplify the script by focusing on basic interactions • Pair students to collaborate on building their code.	• Challenge students to add advanced interactions within the limit • Encourage exploration of random movements or conditional events • Ask students to create and reflect on their strategies for optimising their scripts • Suggest students revisit previous projects and look to optimise their previous code.

Plenary

- “What did you learn about keeping your code efficient?”
- “How did the 20-line limit affect your creativity?”
- “Why is coding efficiently an important skill for game development?”

Assessment questions

How did you design your sprites to make them unique?

What strategies did you use to stay within the 20-line limit?

How does efficient coding improve the functionality or readability of a project?

If you could add 5 more lines, what would you improve or add to your game?