

LESSON 11

Traffic Light Sequence

40 minutes

Coding and Creative Projects

WHAT THIS LESSON DOES

This lesson guides students through creating a traffic light sequence in Scratch. They will learn to create a new project, upload a sprite, understand sequencing, and program the traffic lights to change colours in a specific order.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand and apply the concept of sequencing in computer programming.
- Develop skills in using Scratch to create and modify a project.
- Learn how to upload and utilize sprites in a Scratch project.
- Apply coding skills to create a functioning traffic light sequence.
- Understand and apply the concept of loops in coding to make the traffic light sequence repeat.
- Develop problem-solving skills by modifying the code to change the sequence repetition.

BEFORE THE LESSON

- Open the Scratch site yourself first and confirm the school filter is not blocking it. Have the link ready to share with pupils.
- One device per pupil, or per pair if devices are short. Check pupils can save a file and find it again in a known folder.
- Download the Earth and Homer GIFs supplied with the lesson in advance, or make sure pupils can reach them on their devices.
- For the Explore step, decide whether image searching is allowed on your school network and set a safe search engine if it is.

Traffic Light Sequence

HOW THE LESSON RUNS

Introduction

Open Starter Project

Set it to red

Set it to green

Set it to orange

Make it loop!

Wrap Up

TEACHING MOVES

- 1. Introduction. Ask who has seen a GIF and where. Land the one idea that matters: a GIF is lots of still pictures shown so fast they look like they move.
- 2. Create a new project. Walk through starting a new project once on the board, including deleting the cat sprite, so nobody is left with two sprites by accident.
- 3. Upload the Earth GIF. Have pupils open the Costumes tab and count the 44 costumes. Point out each one is the Earth turned slightly. This is the whole concept made visible.
- 4. Make it 'Animate'. Build the forever next-costume loop with them block by block. Click the green flag together so everyone sees the same spin at once.
- 5. Upload the Homer GIF. Remind them to delete or hide the Earth sprite first, then upload Homer. A pupil coding on the wrong sprite is the commonest snag here.
- 6. Make it 'Animate'. Let them run it fast first so they feel it is too quick, then add wait 0.1 seconds. The problem-then-fix order teaches what the block does.
- 7. Explore. Set clear limits before they search: your chosen search engine, animated GIFs only, sensible topics. Show the change color effect block once, then let them experiment.

WHAT TO WATCH FOR

- Pupils drag the animation code onto the wrong sprite, or leave the old sprite in the project, so nothing moves or the wrong thing moves. Show them the highlighted sprite in the sprite list and that code belongs to whichever sprite is selected. Have them check the right sprite is chosen before coding.
- Pupils expect a single uploaded picture to move on its own and are surprised nothing happens until they add code. Point back to the Costumes tab. The pictures are all there, but the computer only flips through them when the forever next-costume loop tells it to.
- The animation runs so fast it looks like a blur and pupils think it is broken. This is expected before the wait block. Use it as the reason to add wait 0.1 seconds, and let them try different wait values to see the speed change.

DIFFERENTIATION

- Emerging: Sit with them for the first upload and the first loop. Once the Earth is spinning they usually have the pattern. Pair them with a pupil who is one step ahead so they can copy the block order and then explain it back.
- Developing: Have them build the Homer loop from memory rather than reading it off the board, checking against the example only if stuck. Ask them to predict what wait 0.5 seconds will do before they try it, then test.
- Proficient: Point them to the Explore step early: their own GIF, plus a colour effect and a sound. Challenge them to make two sprites animate at different speeds using different wait values, and explain why.