technology

Scratch: loops and events

Lesson at a glance

Project Pitch Bounce on the IWB: a ball that forever-bounces and kicks up on space or a sprite tap. Pupils name what repeated and what started each script, then watch you model the green-flag forever loop and the kick event. In pairs they rebuild one working loop plus one kick event on Scratch, debug until both run, and record on the Investigation Journal what the loop does, what the event does, and one bug or change tried. Close with a short share that revoices loop, event and debug.

Learning objectives

- Use a loop to repeat instructions in a Scratch program
- Use an event (green flag, key press, or sprite tap) to control when something happens
- Debug a program that uses loops or events and record a fix or a change you tried

Before the bell — prep

Check Scratch on class devices (or IWB for turn-taking). Note keyboard vs touch so pairs pick the right kick event. Build Pitch Bounce once before the lesson: green flag → forever → move 10 → if on edge, bounce; plus space or sprite-tap → change y by 40.

Materials

Item	Qty	Per	Source	Low-cost substitute
tablet or laptop with Scratch	1	pair	school devices	run the lesson on the IWB with pupils taking turns

Safety watch-point

Supervise device handling and packing at the end. No other significant hazard — Scratch is on-screen only.

Lesson 27

Teaching moves

- **Getting Started:** Run Pitch Bounce once without naming blocks. Ask what keeps happening and what makes each part start; collect two or three ideas on the board. Keep the full block palette closed until the model step.
- **Loops and events:** Name loop and event from what they saw. Pupils jot one noticed fact in the journal top space only. Hold the full debug definition until pairs hit a stuck program in the build.
- **Model the loop and event:** On the IWB, snap and narrate: when green flag → forever → move 10 → if on edge, bounce; plus space or sprite-tap → change y by 40. For mixed devices, model both kick events. Ask for predictions before you run, then leave the stacks visible while pairs jot the loop and event they will copy.
- **Build, run and debug:** One device per pair if possible. Circulate with prompts: Which block is your event? What is inside the loop? What did you expect? Watch for motion outside the forever, wrong event for the device, missing if-on-edge-bounce, or two forever loops fighting. Success bar is one working loop plus one working kick — not a finished game.
- **Record in your journal:** Paper only, this sitting. Three short notes: what the loop does, what the event does, one bug fixed or change tried. Offer starters: Our loop... / Our event starts when... / The bug was... We fixed it by... Unfinished code with a clear debug note still counts.
- **What the loop and event did:** Take two or three pairs. Revoice with loop, event and debug. Draw out that a loop repeats the inside blocks, an event decides when a script starts, and debugging means change one thing and test again. End with one volunteer naming the difference between a loop and an event.

What it should show

Green flag should keep the ball bouncing around the stage; each kick (space or sprite tap) should jump it up once. If it moves only once, motion is outside the forever. If kick does nothing, the event does not match the device. If the ball vanishes, if-on-edge-bounce is missing. Two forever move scripts will fight each other.

Misconceptions & interventions

- **Pupils think the green flag is the only event, so a key press or sprite tap 'doesn't count' as starting a script.** — Point at both hats on the model: green flag starts the bounce; space or tap starts the kick. Have them run each alone so they see two separate starters.
- **Pupils stack move blocks many times and think that is the same as a loop.** — Hold the forever block beside a long move stack. Show one change inside the loop updates every repeat; the stack needs every block edited.
- **When nothing works, pupils rebuild the whole program instead of changing one thing.** — Stop the pair and ask: change only one block or nest, then run again. That single change-and-test is the debug habit to name in the journal.

Differentiation

Emerging	Developing	Proficient
• Get Script 1 (green flag + forever bounce) working first; add the kick event only when bounce is solid. Pair at the teacher table with the model stacks still visible on the IWB.	• Once both scripts run, name aloud which block is the loop and which is the event, then try one small tweak such as kick size (change y value).	• Add a second kick path (both space and sprite tap) or a second event, still aimed at clear loop-and-event control; note the extra change in the journal.

Cross-curricular hook

Link to English oral language: pairs explain a bug and fix in clear sequence using the starters on the journal page.