Dr Scratch

Summary

Dr Scratch is a free online analytical tool that provides feedback on Scratch (MIT) project progress.

Description

Dr Scratch is a free online analytical tool that evaluates Scratch projects. To use the tool, students insert the URL to their Scratch project, or upload their Scratch project file. The tool analyses the presence of Scratch blocks, functionality of scripts and use of sprites.

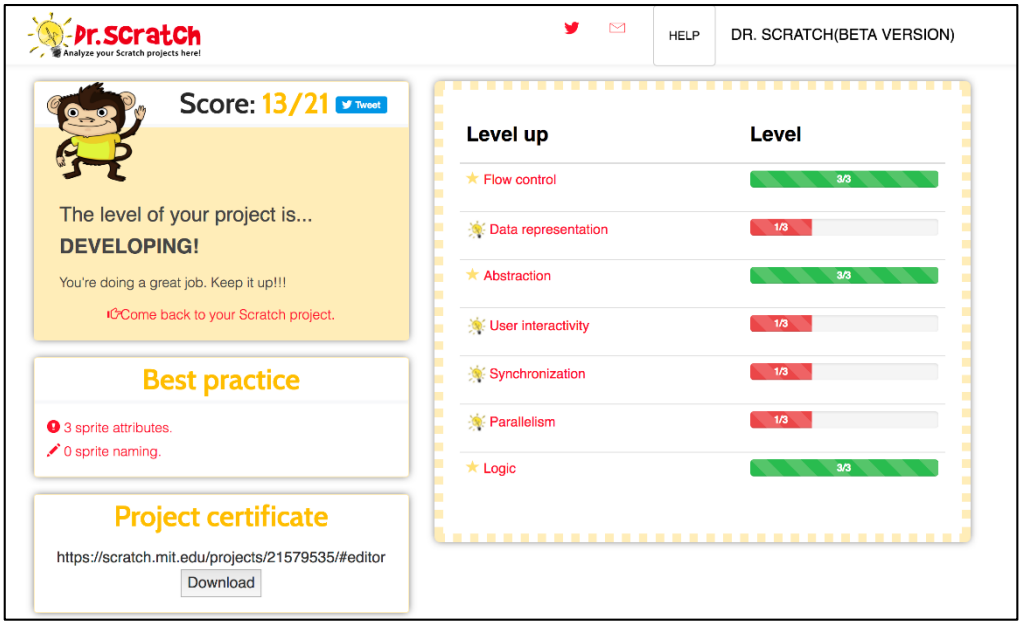
The tool provides feedback on a variety of computational areas, including: flow control, data representation, abstraction, user interactivity, synchronisation, parallelism and logic, as well as use of best visual programming practice (eg use of sprites attributes and naming, and script performance). It provides a score out of 21 and feedback in the form of a progress status (ie 'developing'). Students can print a copy of their certificate, which shows the score, allowing them to document progress over time. However, to capture the report, it is recommended that a screen capture is made.

This tool would be suitable to support formative assessment. It can be used to support students' self-evaluation of their own projects, peer-evaluation or as a support tool for teachers wanting to undertake formative assessment of students' Scratch projects.

Please note that this tool is in Beta phase. It should not be used as a diagnostic tool, rather as something to support formative assessment and drive discussion and self-reflection.

Year level bands

3–4, 5–6



Screen capture of Dr Scratch evaluation summary

Guidance for use

This analytical tool can be used for **formative assessment** of Scratch projects. Prior to using the tool in the classroom, we recommend that teachers review the various indicators that the evaluation report analyses and how students might be guided toward designing more effective programs (eg see the **Logic** page of the Dr Scratch website).

Some recommendations for using this tool are as follows:

- Encourage students to iteratively test their Scratch programs, or plan opportunities throughout a unit for students to evaluate their programs.
- Have students take a screen capture or photo of their evaluation summary so that they can document their progress over time during their project creation.
- Have students compare their first and last reports: What has improved? What has stayed the same? What could they improve next time?
- Record students reflecting on their Scratch progress summary, or ask students to write a reflection piece.
- Prepare for students a blank version of the Scratch evaluation page and have students self-evaluate their projects based on the criteria before using the tool. Invite students to reflect on how accurate their own evaluations were.
- Teachers can use this to guide formative assessment and feedback. Run a student's project through the tool and provide feedback (using the online guide) about how students can improve their projects.

Australian Curriculum Digital Technologies alignment

Years 3–4

Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010)

Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011)

Years 5–6

Design a user interface for a digital system (ACTDIP018)

Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019)

Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020)