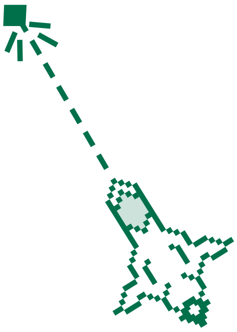
Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking



NIGEL CALDER

describes how *Scratch* can be used to design games to develop mathematical concepts.

Introducing Scratch

Scratch is a media-rich digital environment that utilises a building block command structure to manipulate graphic, audio, and video aspects (Peppler & Kafai, 2006). It incorporates elements of Logo including 'tinkerability' in the programming process (Resnick, 2007). This allows children to combine the programming building blocks (at times incorporating measurements) and to immediately observe the outcome of that programming. The blocks can be taken apart and recombined as the children develop the desired movements and effects. In *Scratch* students use geometric and measurement concepts such as coordinates, angle and length measurements. As well, it facilitates creative problem solving, logical reasoning, and encourages collaboration.

This article examines the ways mathematical thinking emerges when children work with *Scratch*, an interactive, programming language. It describes how a class of Year 6 students used *Scratch* to design an activity for their Year 1 'buddy' class and considers how this facilitated an authentic problem-solving process. The ways their mathematical thinking emerged through this process is outlined, along with further suggestions for using *Scratch* in classroom situations.

Figure 1 shows the *Scratch* workspace with building blocks and actions on the left. The

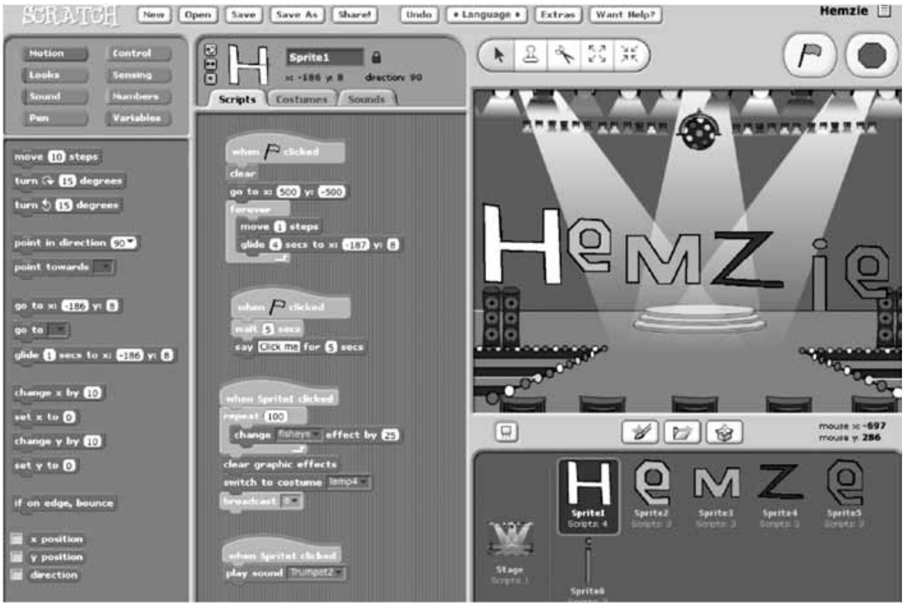


Figure 1. The Scratch workspace.

program with its block of command structure is in the middle, and the stage, where the actions are carried out, is on the right. Below the stage are the icons or photos (sprites) that can be used in the program.

The project

A pilot research project was conducted with a digital-learning class of 26 Year 6 children. Each student had access to their own computer and although this was their first experience with *Scratch*, they were already confident and experienced with a range of software. The students worked in self-selected pairs. Over the two-week research period, the students wrote daily blogs articulating their progress and reflections, students and the teacher were interviewed, and classroom observations (both written and photographic) were recorded. These, along with informal observation and discussion, formed the data.

The first week involved the students doing a range of distinct, structured tasks to familiarise them with the *Scratch* environment. This included creating a dynamic representation of each group's name. An example is shown in Figure 2.

All groups were then given the same design brief: "Design and build a mathematics game suitable for facilitating the number understanding of your Year 1 'buddies'."

The students interviewed their Year 1 'buddy' class partners and consulted the Year 1 teacher regarding appropriate mathematics concepts, and activities with which the class was familiar. This helped determine the types of games they would devise. The younger children also trialed the games partway through the development process and gave formative feedback on how the game worked, what parts they enjoyed, and how it helped their learning.

A feature of the approach taken by the teacher was the sharing of the work that had been done each day. Each project was loaded onto a data stick near the end of the session and one student took responsibility to coordinate displaying the work on the data projector. Each group would explain what they were doing

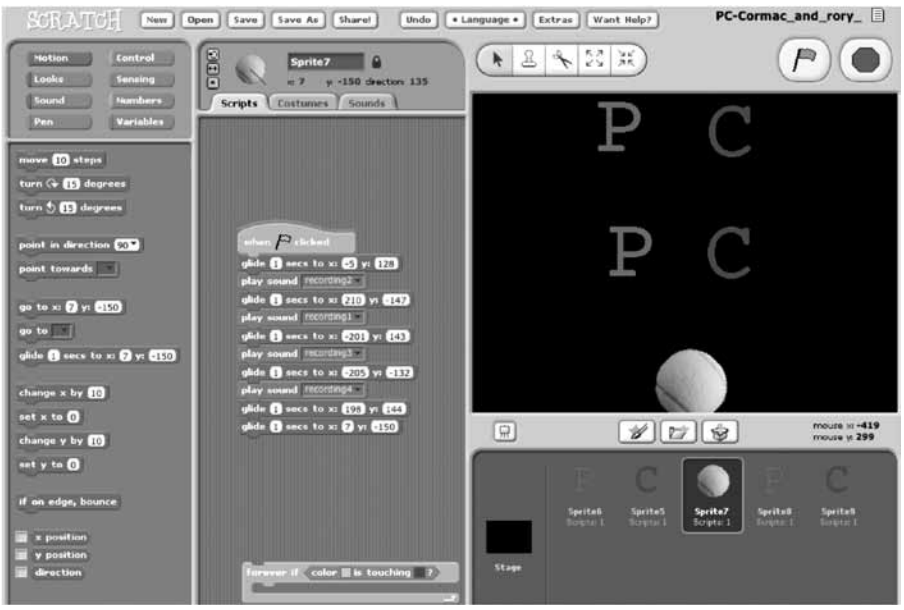


Figure 2. The group PC's workspace, as they animated their group name.

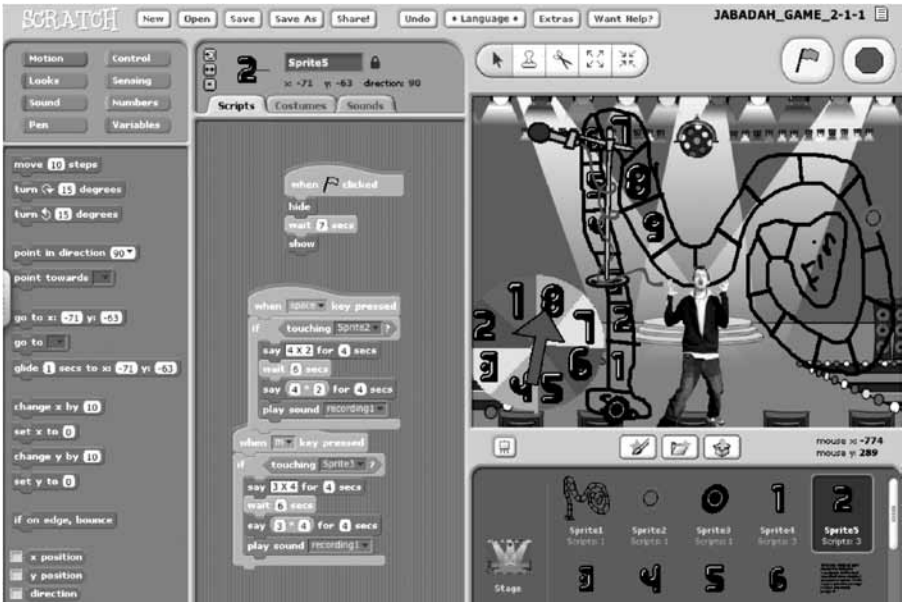


Figure 3. The group 'Jabadah's' workspace as their game is in development.

and any characteristics of their programming. The other students could ask questions and provide feedback and suggestions. The students' respect for each other and confidence with this process was a feature of the classroom culture and clearly had been engendered before the project took place. The feedback session also gave opportunities for the teacher to formatively assess, to identify aspects that might need individual or whole-class feedback, and for students to identify other class members who could assist them with aspects of their design problems.

Mathematical thinking: Using geometry and measurement ideas

The group, Jabadah, (see Figure 3) used geometry and measurement during the process of developing their game. They trialed variations of movement, angle size and coordinates, and linked these to the associated effects. This exploration enriched their understanding of these aspects. Other groups also demonstrated the use of geometric and measurement ideas.

The Jigsaw group explored changing the length of time for repeat movements and varying angle sizes. They later articulated their attempt to make the letters glide into place, eventually figuring out how to use coordinates to specify the movement of the 'sprite' on the stage. The group, Mats, likewise aimed to explore animation and movement. They worked out how to use the glide command and *x*- and *y-coordinates* to move to different positions on their stage. Members of this group described what they were doing as follows:

Stan: We have to remember where the numbers (their 'sprites' for the game they were devising) go, so they all move to the middle and then they mix around to different places.

Matiu: We want to put the numbers in position.

Later, they applied this skill in their game to move asteroids through space. They were observed manoeuvring a spaceship and dodging spinning asteroids. Interestingly, at another point, when writing the script for

Figure 4. The group PC's workspace as their game is in development.

their sprite to move, they initially recorded: “turn 90 degrees, wait one second” 10 times, rather than using a more efficient: “repeat 10 times command.” After choosing a stage and sprite, another group, XE2, was observed immediately engaging with movement and the positioning of their sprite. This involved the use of coordinates to indicate the position to which they wanted the sprite to glide. They were not concerned with the exact position of the coordinates, but more the general position associated with them. They spent time exploring different coordinates and how this affected the position of the sprite, gaining a sense of the relationship between the values of the coordinate and the position on the screen. They also programmed ‘wait time’ of 5 seconds and ‘hide time’ of 6 seconds.

PC also experimented with time, and what the interval signified, when creating their game. They formulated a program that offered simple addition equations such as ‘7 + 4 =’ and the buddy children needed to match the solution to the appropriate number of aliens. One group member, Peter, described how, “If you get this right, it tells you, and then it changes to the next question in 15 seconds.” Figure 4 shows their workspace as their game was developed.

The group, Pig, explored similar areas but with an additional transformation. They wrote on their blog: “We have learnt how to move letters and characters by programming a key on the keyboard to move an object. We learnt that if you use a text box you can’t make an animation with effects, it will just enlarge your sprite.”

They articulated that their initial aim was to find out about position and effects as they were exploring movement and angles.

Programming in *Scratch* appeared to facilitate the children’s understanding of angles and measurement. It allowed them to experiment and explore what was appropriate for their particular context. Errors with programming appeared to have a positive effect in that they prompted further experimentation to achieve the desired movement. The ‘tinkerability’ of *Scratch* facilitated exploration with angles and the measurement of time and length. Students could actively experiment with angle size, for example, in ways that would not be possible without the digital medium. Likewise, the understanding that emerged regarding coordinates was inherent in the process of exploring the movement and position of the sprites. Clements, Sarama, Yelland, and Glass (2008) discussed how game contexts and practice can significantly improve spatial performance. This activity not only involved participants with spatial movement and location while designing the games, the trialing and modification process would also have influenced the children’s spatial awareness.

Problem solving in Scratch

Although the design brief was set within a mathematics context, a central element of the thinking that took place went beyond the area of mathematical problem solving. The students familiarised themselves with the task and then through iterations of action and reflection modified their game. At each juncture, the feedback on their efforts enabled them to modify their approach to re-engage from a fresh perspective. Hence, their thinking evolved and the games became more refined as they reflected on the feedback. The feedback was in various forms: immediate visual feedback within the program as they changed their programming script; feedback from others within the groups; teacher feedback and suggestions; feedback from their buddy class, and from other groups as they viewed each other’s projects.

For instance, the group, Jabadah, began with a “stage” and explored how to move the sprites, and some of the pre-programmed effects. They experimented with moving the sprites that made up the letters of their group name. They wanted to make the J hit the A and set it off spinning, but it moved in a continuous loop. They tried a few options and considered the visual feedback resulting from each change in the coding. They were developing a sense of the relationship between the programming script they had selected and modified, and the associated movement of the sprite on the screen. The next day they continued this experimentation by "using existing scripts to see how to manipulate things differently" (Jabadah group member). During this process they worked out how to design and operate a spinner. When they reached a point of uncertainty they used a 'predict and check' approach, reflecting on the outcome, before refining their evolving script. They said that they "looked at how the different scripts affected the action of the sprites" and "experimented" with the number scripts in their own project by putting in variables and then running the script to see what would happen."

The students' thinking evolved through the problem-solving process. As well as relational thinking, they also used logic and reasoning to evaluate and interpret the situation, before resetting their sub-goals in the investigative process. They generalised from a range of actions and, after reflection, determined the type of command that produced the desired effect. They also responded to other feedback. For example:

MF: How does the code work, tell me what that code means?

James: It just spins randomly and lands on a random place.

Although the question was not answered in the detail intended, the student has reflected on the question and articulated his response in terms of both language of movement (spins, lands) and chance (random). The children also articulated the movement of the spin in mathematical language, which they understood.

Don: All we did was go: "When sprite 15 clicked repeat random 3 to 100 and turn 45 degrees."

Such descriptions demonstrate development in the children's understanding of rotation and the link between the numerical size and the movement of the turn. Data from other groups also highlighted the way *Scratch* facilitated problem solving. For instance, when Geoff had run into a problem with the logic of the scoreboard of their game, he said, "I'll need to problem solve that."

He then investigated spacing, proportion, colour, and size aspects of the scoreboard. The challenge of the problem-solving process was evident in a blog from another group, who said, "We are trying to figure out how to use a gravity effect and how to use the variables. We are finding it challenging to make our character, Jetman, jump in the air without spinning 15 degrees."

The teacher also discussed problem solving in her final interview. She talked about some of the benefits, saying: "The communication and competencies are coming through with the use of it. That whole problem solving and questioning (aspect). So the whole thing of exploratory learning was where it was a very valuable bit of software."

She believed that the children using the program benefited in terms of both problem solving and mathematics.

Conclusions

Scratch software proved to be an engaging and relatively easy-to-use space for problem solving, which at the same time provided a worthwhile and motivating programming environment to explore mathematical concepts. Additionally, it proved to be an effective medium for encouraging communication and collaboration (Otrel-Cass, Forret, & Taylor, 2009). The challenge of creating a mathematical activity or game for younger students overtly positioned the program in mathematics, while implicitly, it simultaneously demanded that mathematical ideas be utilised to develop their game. The children were quickly able to access and understand the programming capabilities and used mathematical thinking in their approach to problem solving. It proved to be a medium whereby programs were easily composed and modified, thus encouraging the use of critical, meta-cognitive and reflective skills. *Scratch* was also intrinsically motivating.

The sharing sessions were pivotal in that they provided a forum for displaying work, and as a way of collectively helping each other to solve programming problems. *Scratch* therefore, provided an opportunity for students to develop their thinking, a key competency that is integral to the New Zealand Curriculum (Ministry of Education, 2007) and also inherent in the proficiency strands of the draft Australian Curriculum: Mathematics. The facilitation of logical thinking from initial empirical concepts, as students test ideas in response to feedback, and the influence of program feedback in the evolution of students' geometric ideas has been reported elsewhere (e.g., Clements et al., 2008).

Although not designed specifically to facilitate conceptual thinking in a particular mathematical area, there were clear indications the children engaged with mathematical ideas. The children's spatial awareness, understanding of angle and time measurements, and sense of position through the use of coordinates, were all engaged to varying degrees. There was also evidence of relational thinking as the children made links between their input, the actions that occurred on screen, and the effect of specific variations of size and occurrence of single or repeated procedures. However, the process the participants undertook also facilitated mathematical thinking by evoking creative problem-solving processes and the development of logic and reasoning as they responded to the various forms of feedback. Most programming tasks in the *Scratch* environment involved the development of mathematical concepts. As well, the Year 1 children used mathematical thinking in number to do the tasks and games that had been programmed for them. Other possible tasks using *Scratch* that would likewise develop mathematical thinking are:

- Design a program that shows the key things you have learnt in our geometry/number/probability (etc.) unit.
- Design a program that moves a sprite in a square, equilateral triangle, rectangle, spiral (etc.) pathway.
- Make an interactive learning object that uses the geometry/number/probability etc. ideas we have been doing in class.
- Design a plan of the classroom.
- Design a program that gets a photo of you dancing in time to a beat.
- Design a program that will allow other students to explore angle size.

Acknowledgement

I wish to acknowledge the rest of the *Scratch* research team: Dr Mike Forret, Dr Kathrin Otrel-Cass, Merilyn Taylor and Sheena Saunders.
Scratch: http://scratch.mit.edu/

References

Clements, D. H., Sarama, J., Yelland, N. J. & Glass, B. (2008). Learning and teaching geometry with computers in the elementary and middle school. In M. K. Heid & G. W. Blume (Eds). *Research on technology and the teaching and learning of mathematics* (Vol. 1, pp. 109–154).

Ministry of Education (2007). *The New Zealand Curriculum.* Wellington: Learning Media.

Otrel-Cass, K., Forret, M. & Taylor, M. (2009). Opportunities and challenges in technology-rich classrooms: Using the Scratch software. *SET: Research Information for Teachers 1*, 49–54.

Peppler, A.P. & Kafai Y.B. (2006). Creative codings: Personal, epistemological, and cultural connections to digital art production. In S. Barab, K. Hay and D. Hickel (Eds.), *Proceedings of the 2006 International Conference of the Learning Sciences.* Bloomington, IN.

Resnick, M. (2007). Sowing the seeds for a more creative society. *Learning & Leading with Technology*, 35(4), 18-22.

Nigel Calder
University of Waikato, New Zealand
<n.calder@waikato.ac.nz>

APMC