Primas

SPIROLATERALS

Abstract

In this lesson, students explore a simple computer microworld. This offers them an opportunity, often rare in Mathematics and Science lessons, to formulate *their own* questions about a phenomenon, and to tackle their questions systematically, using scientific reasoning. The difficulty of the task lies not so much in its content (this makes the task widely accessible), but rather in the opportunity it offers students to experiment, generalise and prove.

Rationale

The task has been used widely in professional development. It offers teachers an opportunity to move away from the traditional one of explaining content to one in which they encourage students to take more active control of the their own learning. The teachers' new role is not simply to stand back and observe, it is rather to encourage students to describe, explain, generalise, prove and communicate as precisely and convincingly as they can.

Discipline

This task is taken from the field of Mathematics. It is not, however dependent on advanced mathematical content and so may be used with both primary and secondary students.

Conceptual aims

As students are able to define their own questions and draw on their own knowledge, the conceptual aims may vary. The situation is geometrical and students may draw on their knowledge of symmetry (reflective and rotational), and algebra when making and proving generalisations.

Age group

The task is suitable for students aged from 9 to 16.

Country

This situation is well-known in England, and has been referenced in many places.

IBL processes involved

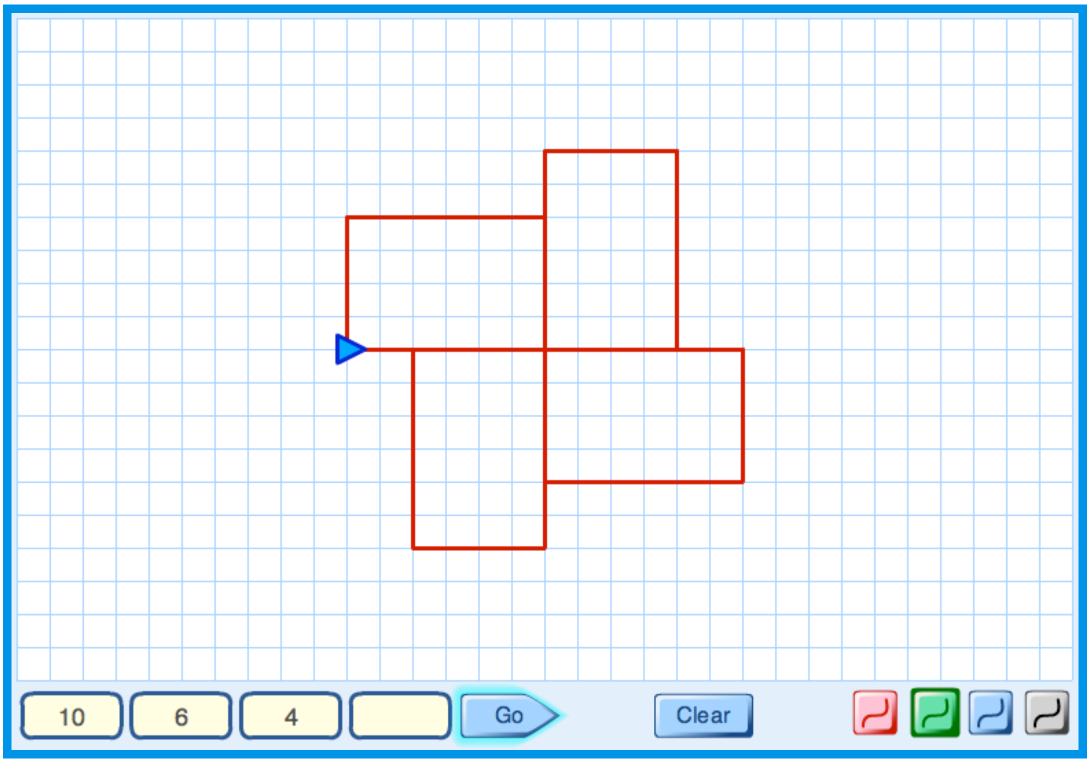
In this task, students:

- Observe a computer microworld, experiment by changing variables, formulate questions and hypotheses.
- Plan a systematic investigation into their own question. They select ways of representing the data they collect, for example, using tables. They draw on their mathematical knowledge to describe their findings (e.g. symmetry).
- Students carry out their investigation, systematically collecting, documenting and analysing data and further information.
- Make, explain and prove generalisations (for example, entering a sequence of three numbers will result in a closed shape)
- Communicate results and reflect on what has been learned.

Links and references

This task and the software that accompanies it was designed by Malcolm Swan and Daniel Pead and form part of the Bowland Maths professional development resources that have been developed in England and distributed to schools across the UK. They are reproduced here with kind permission of the Bowland Charitable Trust.

Spirolaterals



Type some numbers into the *Spirolaterals* machine.
Press "Go" and watch what happens.

- How do the numbers control what is drawn on the screen?

- List some questions to explore.
  These might start like this:
  "How can we make the computer draw ....?"
  "What will happen if we .....?"

  Try and answer your own questions!

- Make up some conjectures. These might start like this:
  " When you use three numbers then ..."
  " When you repeat a number then ..."
  " If you change the order of the numbers then ..."

  Can you *prove* your conjectures?

A possible lesson plan

Lesson duration

This lesson should last between one and two hours. The timings below are only approximate, and will vary from class to class.

Preparation and Resources needed

Each pair of students will need:

- access to a computer computer loaded with the applet, *Spirolaterals*.
- some 1 cm squared paper to record their ideas.
- a clipboard to lean on (if computers take up the table space)
- pencils and rulers.

It is also helpful to have a data projector in order to present the task to the whole class, and to provide an opportiunity for class discussion about the problem.

Introduce the task to the class 5 minutes

Give each student a copy of the problem handout and explain the purpose of the lesson:

The aim of today's lesson is for you to explore a simple computer program.
Your task is to try to answer these questions:

What does the program do?
What interesting problems does it suggest to you?
Can you solve any of these problems?
Can you make any conjectures and prove them?

Explain how students are expected to work:

I want you to work in pairs to see how the software works.
Try to record what happens when you enter different numbers/ press different buttons.
As you do this, begin to think of some possible problems to investigate.
You might, for example make up problems that start with the words:
"How can we make the computer draw ....?"
"What will happen if we .....?"
I'll ask you to share your ideas for problems with the rest of the class in five minutes!

Issue students with clipboards, 1 cm squared paper, pencils, and rulers.

Students explore the microworld and generate problems 15 minutes

Allow students 5 minutes to explore what happens as they type numbers into the software.
Go round, encouraging students to describe what is happening precisely.

Tell me how the computer knows what to draw when you enter those numbers.

If I enter these three numbers (1, 2, 3) what will it draw when I press go?
What is the computer doing with these three numbers?

Encourage students to describe what is happening as clearly and fully as they can.
For example, they might say:

You start by facing to the right.
You move 1 unit forward then turn left 90°
You move 2 units forward then turn left 90°
You move 3 units forward then turn left 90°.
You move 1 unit forward then turn left 90°.
... and so on repeating 1,2,3 until you get back to the start.

Repeat this process with other numbers until you think students know how the program draws the shapes.

Next, discuss possible problems to explore and list some on the board.
Students might suggest ideas such as the following:

What will happen if we enter a single number and press "Go".
2 numbers? 3 numbers? 4 numbers?
Can we predict the types of shapes we will get?

What happens when we change the order of the numbers?
So how is (1,2,3) different from (1,3,2)?

Do the shapes always go back to the start?
When do they? When don't they?
How can we predict this from the numbers?

When do the shapes have rotational symmetry?
Can we predict this from the numbers?

When do the shapes have line symmetry?
Can we predict this from the numbers?

What happens if we enter the same number more than once?
What happens with 3 numbers, like (1,1,2); (3,2,3)?
What about 4 numbers, like (1,3,4,4)...?

Ask students to choose a particular problem to work on. Encourage them to be systematic as they try to answer their problem. Discuss how they should record their work.

When you think you have some hypotheses or conjectures, I want you to be able to show me the evidence for this. So as you work, try to keep careful notes of what you try. You may like to copy some of the diagrams by taking screen shots and pasting them into a word processor.

Students work on their problem 20 minutes

As students work on the problem, prompt them to think strategically and analytically:

> *Can you state your problem to me clearly?*
> *What examples have you tried so far?*
> *What are you keeping fixed? What are you changing?*
> *Can you do this in a systematic way?*
>
> *What have you found out so far?*
> *Can you see any patterns or relationships here?*
> *Can you explain **why** your idea seems to work?*
>
> *How are you keeping a record of your work?*
> *Can you use a helpful notation?*
> *Why do you need to do this?*
> *Can you use the computer to do this?*

Students report back and share their findings 20 minutes

When most students have made significant progress with their problem, invite a few pairs to come to the front and communicate their ideas to the rest of the class. They can show some patterns on the software itself, projected for others to see. It does not matter if some have not yet reached any conclusions. They can still share their approaches and ideas.

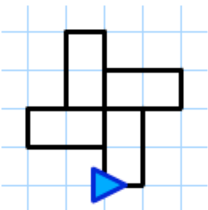
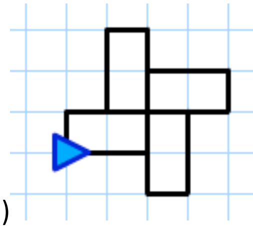
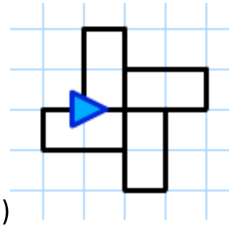
> *Let's stop and share some of the different approaches we have used and consider what you have discovered.*
>
> *Tell us about:*
> - *the problem you are solving;*
> - *how you have organised your work;*
> - *any conclusions you have reached so far.*
> - *any explanations you have for your answers?*
>
> *As students share their ideas, ask others to contribute suggestions, further examples or counterexamples and ideas of what to do next.*

For homework, students could be asked to write an account of their discoveries.

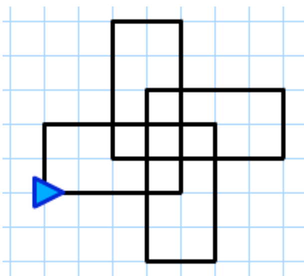
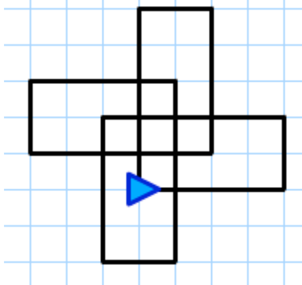
Discoveries students might make

This is an example of a microworld that only takes a few seconds to learn how to use, but is rich in mathematical possibilities. The computer version allows you to explore the relationship between the input numbers and the diagrams obtained.
Pupils may discover many things empirically. Explaining and proving *why* these patterns work is of course an important, more advanced challenge:

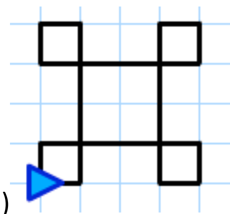
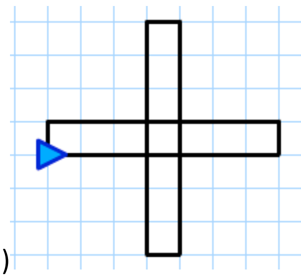
- When a single number *a* is entered, a square with side length *a* is obtained.
- When two numbers *a, b* are entered, a rectangle with sides *a, b* is obtained:
- Any cyclic reordering of the three numbers: *(a,b,c); (b,c,a); (c,a,b)* gives the same design, but start in different places:

(1,2,3)

(2,3,1)

(3,1,2)


- Changing the cyclic order of the three numbers *(a,b,c)* to *(a,c,b)* produces a reflection:

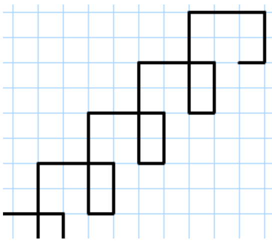
(4,5,2)

(4,2,5)


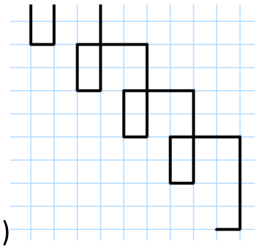
- Three numbers where one or more is repeated *(a,b,b)* or *(a,b,a)* give patterns with lines of symmetry:

(1,4,1)

(4,4,1)


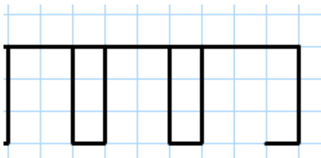
- Patterns with 4 different numbers *(a,b,c,d)* tend to "walk off" the screen...
  - if a>c, then walk is to the right;
    if a<c, then walk is to the left;
  - if b>d, then walk is upwards;
    if b<d, then walk is downwards;
  - if one pair are equal, a=c or b=d then the walk is vertical or horizontal, respectively.
  - if a=c and b=d the we get a rectangle.

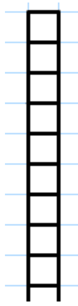
So, (1,2,3,4) is to the left and downwards; (1,4,3,2) is to the left and upwards;

(1,2,3,4) 

(1,4,3,2) 

(1,3,4,3) is horizontal to the left; (1,3,1,4) is vertical downwards;

(1,3,4,3) 

(1,3,1,4) 

(2,3,2,3) is a rectangle:

(2,3,2,3)