

This exercise is meant to help you get used to the Occam programming language and the Browser Occam website. For a full reference on Occam, see this page (tba).

The main feature of occam is channels. *explain channels*

1. *Sequencing.* Here is a program that allows you to move a red cursor left and right across the top of the screen using 1 and 2, and draw dots using 3. We draw to the screen using the **GRAPHICS** channel, but we'll also store what we drew in our own **pixelrow** array.

code

Here is an extension to the program, which will copy everything you drew downwards. The end result should look like vertical stripes.

code

Combine this with the previous program using the **SEQ** command, so it runs after you finish drawing. How will you signal that you have finished?

2. *Parallelising.* Now we will explore parallel processes. We have been drawing each row in order from left to right. Instead, let's make each pixel the job of a single worker. When the worker is done with one pixel, it will move down to the pixel underneath, until reaching the bottom of the page. You'll need to use the **PAR** command.

You should find that the image is now drawn from top to bottom, rather than row by row.

3. *Channels.* Adapt the program so that each pixel beneath your drawing is the maximum of the colours above and to the left of it. The end result should look like a skewed mountain range.

In order to do this, your workers will need to communicate with each other. Each one needs to know the colour to its left, and to communicate its own colour to the right. For this, use a *ring* of channels (explain the ring).

4. *Order of events.* In what order are the pixels of the image drawn now? Is it completely random? Why or why not?

5. *Cellular automata.* Read about one-dimensional cellular automata on **Wolfram Mathworld**. Pick a rule you like, and see if you can adapt your program so that it draws each row of pixels according to the rule.