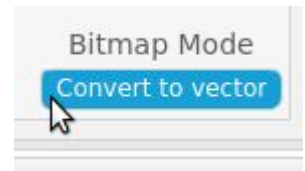


## Week 2

Last week we looked at a programming puzzle game called SpaceChem. That game has two robots which move around a grid of 'cells'. This week we will see how to make our own grid in Scratch, and start thinking about what would make a good game to play on that grid. Start a new Scratch project, and delete the cat sprite.

### The 'cell' sprite

First draw a 'square' sprite. This is best done in 'vector' drawing mode, chosen at the bottom-right of the drawing window:




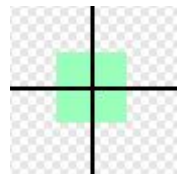
Then use the 'rectangle' tool  with 'filled in' chosen



, and the 'shift' key held down to draw a square near the centre of the drawing area. Choose a colour for it.

Make sure the 'costume centre' is somewhere near the middle of the square, by

using the  tool and clicking near the centre of the square you've drawn:



### Clones

We will need lots of squares in our grid. One way would be to duplicate our square sprite lots of times. This would work, but every time we had to make a change to its scripts, we would regret it, because we'd have to do the same work fifty times.

So instead we will use the 'clone' powers of Scratch. You can read more about the idea of 'cloning' on the [Scratch Wiki page](https://scratch-wiki.org/wiki/Cloning). Cloning will create many copies of the same sprite.

### One row of squares

Let's say we want one row of our robot's grid to have 8 cells in it. We can start off by making 8 clones when the green flag is clicked.

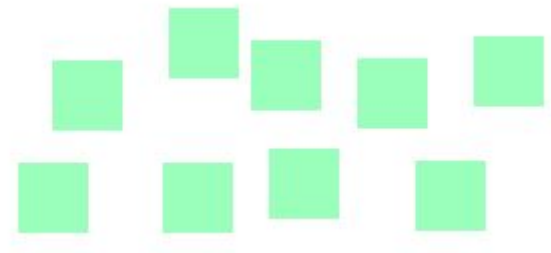
**Add this script to the cell:**

(The 'create clone of myself' block is in the 'Control' section.)

But if you hit 'green flag', it will look like just one new copy, because all the clones will be exactly on top of each other.



You can move them around with the mouse to check there really are eight there (plus the original one):



## Coordinates

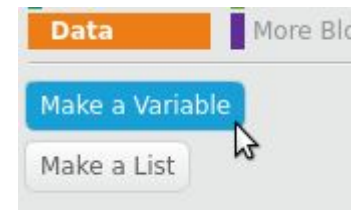
But to make a grid, each square needs to be in the right place. Scratch gives you a [coordinate system](#) to describe exactly where something is on the stage. A sprite's 'x' number says how far right of the centre it is. Its 'y' number says how far above the centre it is. A negative 'x' number means 'left of centre'. A negative 'y' number means 'below centre'.

So we want to spread our first row of squares out horizontally, by giving them 'x' values which increase.

## Each cell needs to know where in the row it is

Each square will need to know which one it is in the row, by remembering its number along the row. In programming we often start counting at zero, so the 8 squares will be '0', '1', '2', ..., '7'.

To remember its position along the row, a square needs a variable just for itself, not shared with the rest of our Scratch program. To create the variable, choose the 'data' group of blocks, and click 'Make a variable':



In the window which appears, call your new variable 'cell-x', and make sure you choose 'For this sprite only', as shown at the right:

When you need a variable, there are two things you need to think about:

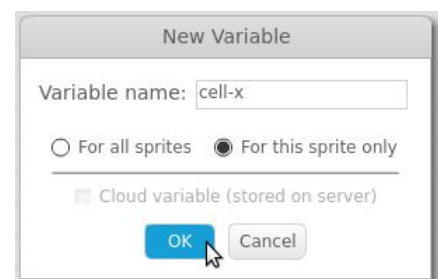
What should it start off being?

When and how should it change?

For making a row of cells, the 'cell-x' variable needs to start at zero, because that's the value we want the leftmost cell to have, and then each time we make a clone, we want 'cell-x' to count up one. Each *clone* gets its own copy of this variable, with the right value in it.

**Change the script to what's shown here:**

But we *still* see all the clones appearing on top of each other, because we're not using this new variable anywhere.



## Moving each cloned cell to the right place

Scratch lets you run a stack of blocks when a new clone is created, by using this 'hat' block, from the 'Control' section.



When we've made a new square clone, we want to tell it to move to its correct coordinates in the row. Each square needs to be some distance — let's guess 20 — to the right of the previous one, so we need to add 20 to one clone's 'x' number to get the next clone's 'x' number. And adding up some number of '20's is the same as multiplying, so we can work out the correct 'x' number for square number '3' by

$$\text{'x' number of square number 3} = 20 \times 3$$

We will use our new variable 'cell-x' instead of '3', to make this work for any clone.

In Scratch, and many other programming languages, the 'star' character \* is used to mean 'multiply'. (Although we're not using it here, the 'slash' character / is used for division.)

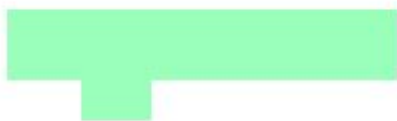
For the 'y' number, we'll start by putting the row half-way up the stage, which has 'y' number zero.



**Add this script to the cell:**

If you run this, it does something, but we're not quite there.

Mine looked like this:



There's the original square, with 8 clones on top of it, but all overlapping, so they look like one big rectangle.

(If your squares are bigger, they might end up bunched up against the right-hand edge of the stage. This is because Scratch won't let you move

a sprite totally off-stage. If this is happening, the changes we're about to make will fix it, so carry on anyway.)

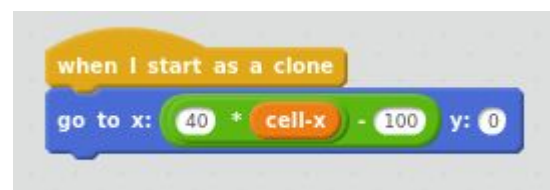
My guess of '20' in 'each square should be 20 along from the previous one' wasn't big enough. Experiment with changing the '20' until you get a pattern you like. For the square I drew, I liked how spread out it looks with 40. **Adjust the number in the 'clone start' script:**



Except it's still not OK because the pattern crashes into the right-hand edge of the stage.

We need to move the whole pattern left. Moving something left is the same as making its 'x' number smaller, so we'll subtract (let's say) 100 from the 'x' we're telling it to move to.

**Change the script by using a subtract block:**



The old  $(40 \times \text{cell-x})$  is now  $((40 \times \text{cell-x}) - 100)$ .

Clicking green-flag now gives:

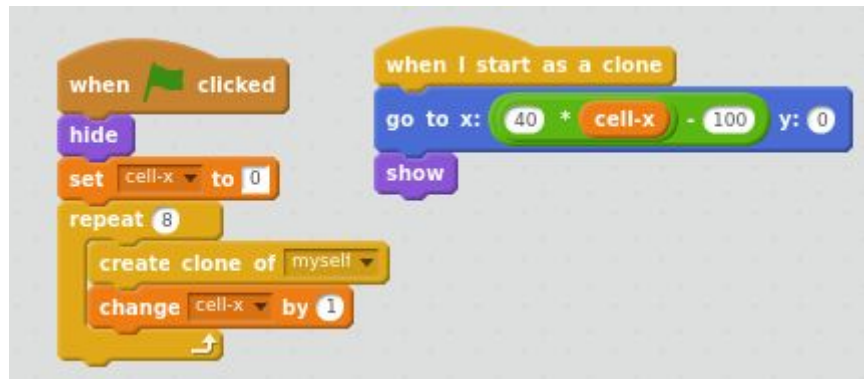
Experiment with different numbers for the '40' and '100' until you like the look of it.



## Hide the original square

The row is cluttered by the original square in the middle of it. Let's 'hide' it; but then we need to 'show' each clone as it's created.

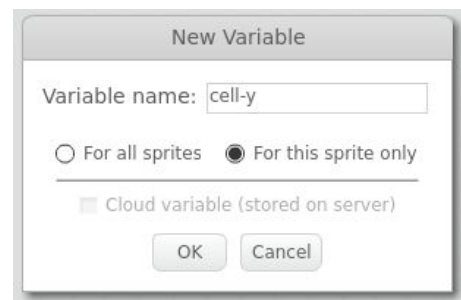
**Change the scripts by adding 'hide' and 'show':**



## A grid is made up of lots of rows

We've now done most of the thinking about how to create lots of copies of the square. It should be quite similar to get a grid made of lots of rows. So we'll go through this a bit more quickly.

Each square needs a 'this sprite only' variable to remember which row it's part of, so **make a new variable**:



And we need to set this variable's starting value, and change it, 'repeating' our row-making blocks six times.

**Change the 'green flag' script as shown to the right:**

(The blocks we had a minute ago go inside the 'repeat 6'.)

And we need to do a similar calculation to work out the 'y' coordinate where we want each clone to go to.

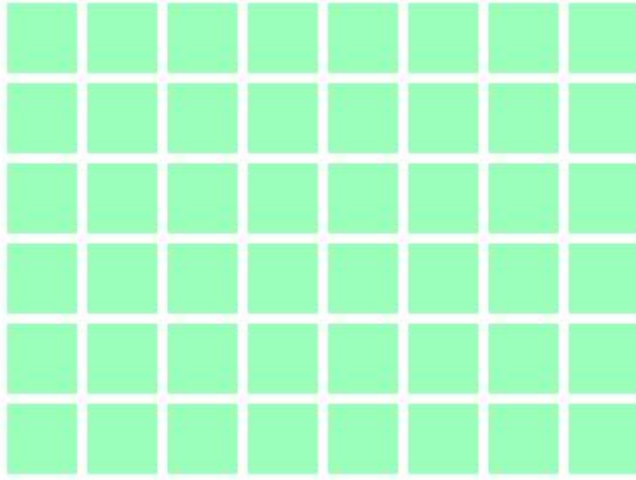
**Change the 'clone start' script:**



(The 'y' part is new.)



You should now have your grid!



## Key points

Using clones to get more than one of a sprite.

'For this sprite only' variables, so each clone has its own copy of a variable.

Using coordinates with arithmetic to work out where to put something.