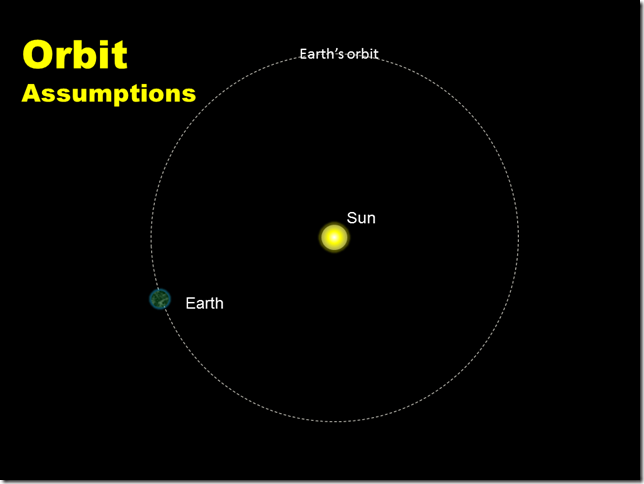
# The orbit lesson

## Teacher guide

**Step 1: Abstraction—what’s in the model?**

Use slide 2 of the presentation as a prompt to quiz the whole-class on their knowledge of the Earth’s orbit of the Sun.

[](http://www.timeaglestone.co.uk/wp-content/uploads/2014/05/Slide2.png)

Some things the kids might come up with:

* the orbit isn’t really circular
* the Earth spins as it rotates
* the Moon!
* the scale of the picture isn’t accurate
* we are only looking in two dimensions

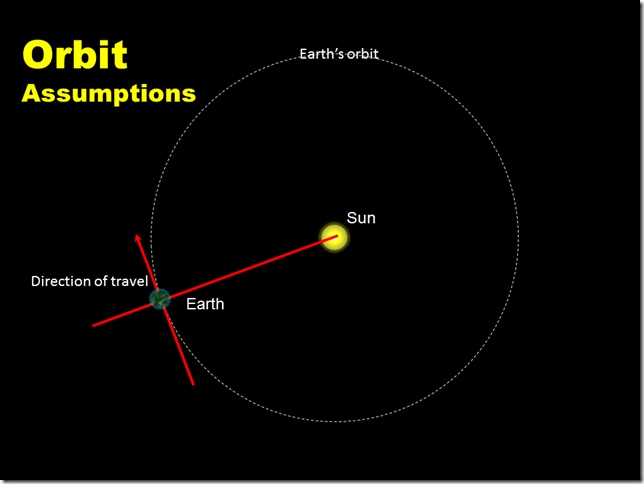
The learning objective for the lesson is to work out how we could model an orbit on the computer. Decided on what the important things are that we needed to consider in order to build this simple model.

**Step 2: Abstraction—how does it move?**

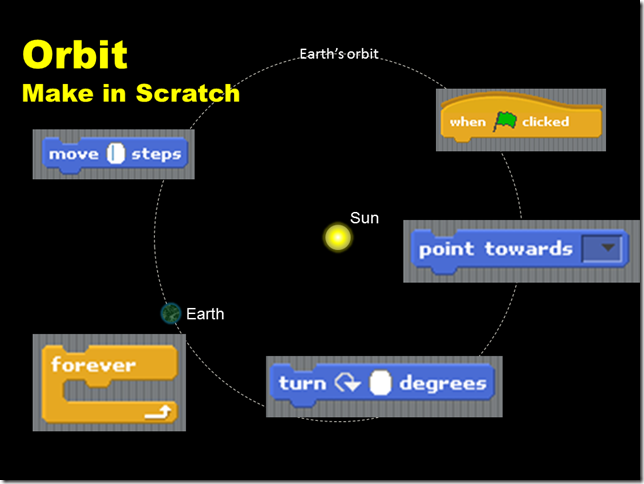
We need some way of making the Earth move. Again, do this through questioning, albeit in a more directed way.

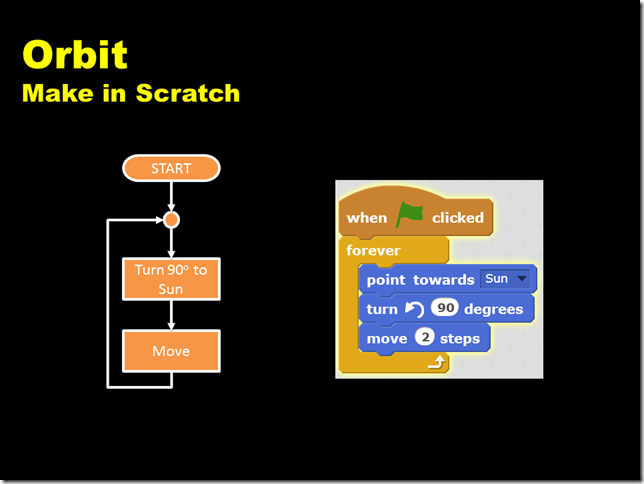
Explore the direction of travel of the Earth at any one moment in time. First guesses tended to be that the Earth was moving towards or away from the Sun—some have the notion that the Sun was attracting the Earth, but avoid discussing acceleration under gravity and focus on how we could practically build a model.

Look at slide 3 and decid that, at any one point in time, and for the purposes of our 2D model, the Earth is moving at a right-angle to the Sun.

[](http://www.timeaglestone.co.uk/wp-content/uploads/2014/05/Slide3.png)

**Step 3: A simple algorithm—how could we make it?**

The next slide incrementally introduces the Scratch elements we could use, and they quickly spot how we could arrange the point, turn and move commands.[](http://www.timeaglestone.co.uk/wp-content/uploads/2014/05/Slide5.png)

We then look at the sequence, wrapped it in a loop and referred to a simple flow-chart:[](http://www.timeaglestone.co.uk/wp-content/uploads/2014/05/Slide6.png)

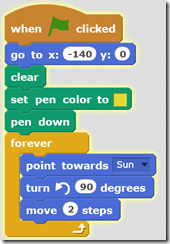
**Step 4: Code it!**

Make the model in Scratch.

This bit is straightforward and all should get a simple orbit working.

Stop at a couple of points during the practical stage and extend the model. Key ideas here are code re-use: make a Mercury and a Venus by duplicating the Earth and changing the start positions by dragging the new planets into position (also change the picture).

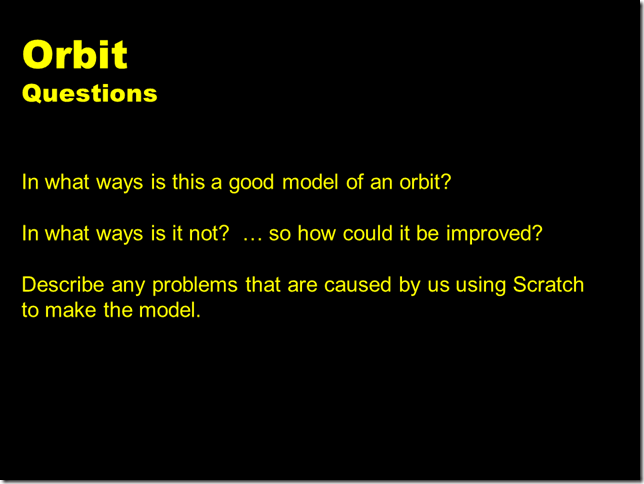
Having more than one planet shows a problem with the model: the orbits tend to spiral out until they reached a stable distance. Used the **pen-down** command to trace the paths of the planets and see the spirals. This makes a good discussion point for the plenary about shortcoming of the model. (The spiral comes from a limitation of Scratch—the ‘screen’ is quite small in Scratch and the turn 90 degrees loses accuracy as does the moving and integer number of steps.)

[[](http://www.timeaglestone.co.uk/wp-content/uploads/2014/05/orbit-script-earth.png)](http://www.timeaglestone.co.uk/wp-content/uploads/2014/05/orbit-script-earth.png)This also made some students add start positions to the planets (all good stuff as we were starting to consider the encapsulation of data). Their final program looked something like the snippet on the left.

Some students made a moon for the Earth. The pattern it traced was interesting. It also bumped around the edge of the screen giving us something else to talk about at the end of the lesson.

By the end of the practical session, most students had developed their models well beyond the original specification.

**Step 5: Plenary**

We used the questions on the final slide as a prompt for reflection.  
[](http://www.timeaglestone.co.uk/wp-content/uploads/2014/05/Slide7.png)

Again, some good comments came back from the class that linked nicely with our original discussion on assumptions.

I was also pleased that many were able to offer good suggestions about some of the limitations arising from Scratch; it’s not that Scratch is poor, it’s that any tool being used to implement something will come with constraints.