

Math on a Sphere

Topic 1: A "Turtle" on a Sphere

If we're going to make artistic or mathematical patterns on the sphere, we need to have a pen to draw with. In our system, the pen is called a *turtle*. The idea here is that we imagine a little turtle-like creature moving around on the surface of the sphere, drawing a line as it goes. By telling the turtle how to move on the sphere, we can draw lines and make patterns.

We'll begin by showing how the turtle can draw narrow white lines on the sphere; very soon we'll see how to change the color and thickness of those lines. With that in mind, let's start with the beginning screen:



Figure 1. The starting screen.

Now, move your mouse into the bottom left window—the one labeled "Command Line Window"—and type the following command (with a line return at the end):

```
forward 90
```

The window should now look like the figure below:



Figure 2. The turtle has drawn a line from the equator to the North Pole.

What we've just done is move the turtle from its starting position on the sphere 90 "steps" forward. It's convenient for us to think of the turtle's starting position as being on the "equator" of the sphere; so after the `forward 90` move, the turtle is now sitting right at the North Pole.

As things now stand, we can't quite *see* the turtle anymore on the screen, because of our viewing angle on the sphere. If you want to see the turtle, you can move the mouse arrow over to the window at the right of the screen and (with a little experimentation) learn to move the sphere about interactively. Try "grabbing" the sphere with the mouse and turning it so that you can see the turtle at the North Pole, more or less as in the figure below.

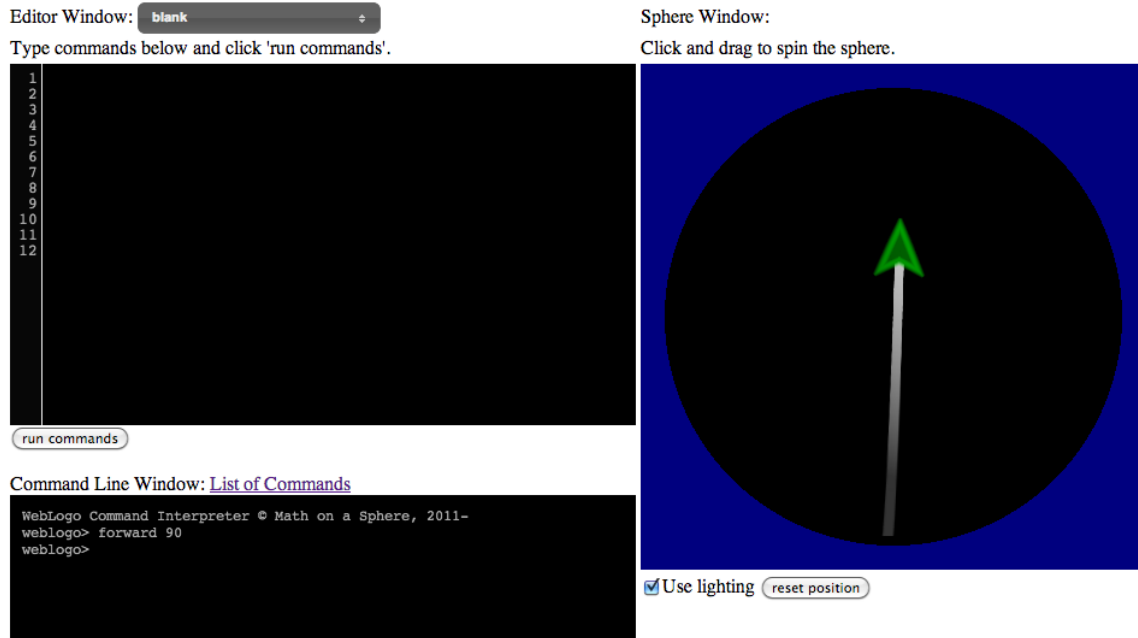


Figure 3. Using the mouse to "turn" the sphere so that we can see the turtle at the North Pole.

If you want to return the view of the sphere back to where you started, you can use the mouse to select the "reset position" button directly under the sphere window. Now the screen will look the way it did in Figure 2, with the turtle nearly invisible toward the top of the sphere view.

Let's do two more steps before we finish with this first topic. First, let's turn the turtle around by making it do an "about-face". To do this, we will need to turn the turtle 180 degrees around its center. The way to do this is to move the mouse back to the Command Line window, and type the following command:

```
right 180
```

Once you do this, the turtle will have turned ("counterclockwise", toward its right) so that it is facing back along the line that it just drew. Now type the following command:

```
forward 90
```

The turtle will now be back in its original position, but facing "downward", toward the South Pole, as in the figure below.

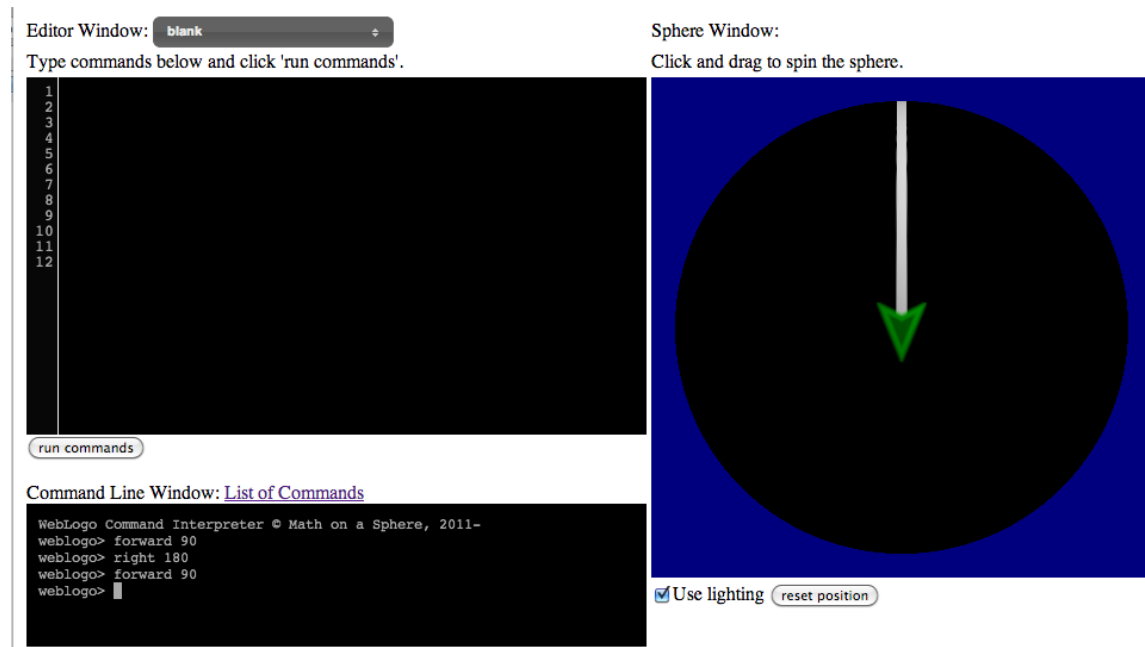


Figure 4. The turtle back in its starting position (but with the opposite heading).

We can turn the turtle back around to its starting heading by making it do one more 180-degree turn; let's do that by typing another right turn command:

```
right 180
```

Now the turtle should be facing back toward the North Pole.

One final command to learn in this first exploration: let's "clear out" the line that we drew. To do this, we can type the following command:

```
ca
```

Here, "ca" stands for "clear all". This is our way of erasing everything on the sphere and bringing the turtle back to its starting position and heading. Once we've typed this command, the sphere will look as it did at the beginning:



Figure 5. Clearing our drawing (and resetting the turtle to its original position).

In general, the `ca` command "erases" all the lines that we've drawn and returns the turtle back to where it started. In this particular example, when we typed `ca`, the turtle was *already* in its starting position, so it didn't have to move back.

This completes our first exploration in Math on a Sphere; we can now type commands to move the turtle on the sphere, turn it in place, and "zero out" our work by clearing the sphere and returning the turtle to its starting position. In the next exploration, we'll experiment with the `forward` and `right` commands.