

# Coordinate Systems for NetLogo 3D, Version 2.1pre1, Take 1

Ralph Abraham

May 1, 2005

## 1 The World

The three-dimensional world, in mathematical abstraction, is Euclidean three-space,  $\mathbf{R}^3$ , with real number coordinates, say  $(u, v, w)$ . In computer simulation, the coordinates may take only discrete values, normally floats or doubles. Also, a bounded world box is defined by bounds:

$$u_{min}, u_{max}, v_{min}, v_{max}, w_{min}, w_{max}.$$

For example, a closed box may be specified as,

$$\begin{aligned}u_{min} &\leq u \leq u_{max} \\v_{min} &\leq v \leq v_{max} \\w_{min} &\leq w \leq w_{max}\end{aligned}$$

A three-dimensional torus,  $\mathbf{T}^3$ , may be specified in the same way, but with the identification of  $u_{min}$  with  $u_{max}$ ,  $v_{min}$  with  $v_{max}$ , and,  $w_{min}$  with  $w_{max}$ .

## 2 Screen Coordinates, the 2D Graphics Window, the 3D Graphics Display, and the 3D Graphics Box

In NetLogo 3D, there are two graphic displays. There is a 2D Graphics Window, as in NetLogo 2D, and an auxiliary 3D Graphics Display. In the 2D Graphics Window one may see the turtles as usual. But each turtle has, in addition to the visible screen coordinates,  $(xcor, ycor)$ , an additional attribute,  $zcor$ , which is a compulsory turtle variable. In the auxiliary 3D Graphics Display, this additional attribute functions as a third spatial coordinate, and the turtles may be seen distributed in the 3D Graphics Display according to their values of all three turtle variables  $(xcor, ycor, zcor)$ . We refer to this 3D coordinate domain as the *3D Graphics Box*. It is outlined in white within the 3D Graphics Display.

As in NetLogo 2D, there are two coordinate systems for the 3D Graphics Box. In addition to the three turtle coordinates,  $(xcor, ycor, zcor)$ , there is also an integer-valued system,  $(pxcor, pycor, pzcor)$ ,

for patches (3D cells). Both systems are toroidal, that is, wrapped around the faces of the 3D Graphics Box.

For patches, we have bounds given by the variables, *screen-edge-x*, *screen-edge-y*, and *screen-edge-z*, which are attributes of the 2D Graphics Window widget. That is,

$$\begin{aligned} -\text{screen-edge-x} &\leq \text{pxcor} \leq \text{screen-edge-x} \\ -\text{screen-edge-y} &\leq \text{pycor} \leq \text{screen-edge-y} \\ -\text{screen-edge-z} &\leq \text{pzcor} \leq \text{screen-edge-z} \end{aligned}$$

The default values are *screen-edge-x* = *screen-edge-y* = *screen-edge-z* = 17. Hence the 2D Graphics Window is  $17 + 1 + 17 = 35$  patches square, with also a depth of 35 in the *z* direction, visible only in the supplementary 3D Graphics Display. The third dimension is perpendicular to the (*xcor*, *ycor*) grid of the 2D Graphics Window, increasing from the back to the front, so as to comprise a right-handed 3D coordinate system.

The bounds may be set, as in NetLogo 2D, by editing the 2D Graphics Window widget. Clicking the "More" button, one accesses a dialog box for setting the *pxcor*, *pycor*, and *pzcor* bounds, as well as the size of a voxel, which must be cubical.

Because this system is toroidal, asking the patch with *pxcor* = *screen-edge-x* + 1 to do something results in an action by the patch at the left edge of the screen, and similarly with the *pycor* and *pzcor* coordinates.

Turtle coordinates are also toroidal, as follows. Let us suppose, for simplicity, that the 2D Graphics Window has the default bounds, 17. Then a turtle with *xcor* = 17.0 appears in the center of a patch at the right edge of the screen, while a turtle with *xcor* = -17.0 appears in the center of a patch at the left edge of the screen.

Similarly, a turtle with *xcor* = 17.499 appears at the right edge of the screen, but moving it microscopically to the right, to *xcor* = 17.500, will jump it way around to the left edge of the screen. And a turtle with *xcor* = -17.500 appears at the exact left edge of the screen, but moving it microscopically to the left, to *xcor* = -17.501, will jump it way around to the right edge of the screen. The behavior of a turtle's *ycor* and *zcor* are similar. The 3D Graphics Box is closed at the West, South, and Back faces, and open at the East, North, and Front faces.

The toroidal behavior, from the mathematical point of view, is a mapping from the Euclidean space,  $R^3$ , to the three-torus,  $T^3$ , by division, or reduction modulo the action of a group. In the following, we will indicate the Euclidean coordinates by (*x*, *y*, *z*), so the reduction map is,

$$R^3 \rightarrow T^3; (x, y, z) \mapsto (\text{xcor}, \text{ycor}, \text{zcor})$$

### 3 Affine Isomorphisms

We now wish to relate the world coordinates, (*u*, *v*, *w*), and the box coordinates for turtles, (*xcor*, *ycor*, *zcor*). As the world coordinates are Euclidean, while the turtle screen coordinates are toroidal, we will

describe directly only the affine isomorphism between Euclidean world coordinates,  $(u, v, w)$ , and Euclidean turtle-screen coordinates,  $(x, y, z)$ , or,

$$R^3 \rightarrow R^3; (u, v, w) \mapsto (x, y, z)$$

This is defined by the equations,

$$\frac{x - x_{min}}{x_{max} - x_{min}} = \frac{u - u_{min}}{u_{max} - u_{min}}$$

$$\frac{y - y_{min}}{y_{max} - y_{min}} = \frac{v - v_{min}}{v_{max} - v_{min}}$$

$$\frac{z - z_{min}}{z_{max} - z_{min}} = \frac{w - w_{min}}{w_{max} - w_{min}}$$

or, solving for  $(x, y, z)$  in terms of  $(u, v, w)$ , we have,

$$x = (u - u_{min})(\Delta x / \Delta u) + x_{min}$$

$$y = (v - v_{min})(\Delta y / \Delta v) + y_{min}$$

$$z = (w - w_{min})(\Delta z / \Delta w) + z_{min}$$

where  $\Delta x = x_{max} - x_{min}$ , and similarly for  $u, y, v, z$ , and  $w$ . Similar equations determine  $u, v, w$  in terms of  $x, y, z$ .

## 4 Perspective

The Euclidean space of coordinates,  $(x, y, z)$ , or rather the 3D Graphics Box as fundamental domain of the toroidal coordinates,  $(x_{cor}, y_{cor}, z_{cor})$ , is imaged into the 3D Graphics Display by means of a perspective transformation. NetLogo 3D uses a monocular observer metaphor for this operation, which we now summarize in  $(x, y, z)$  space.

The viewer looks from a location in  $(x, y, z)$  space, the eye point,  $E$ , toward another point, the gaze point,  $G$ . An image plane (two-dimensional affine subspace),  $I$ , is defined, orthogonal to the ray from  $E$  to  $G$ , at the point  $F$  on that ray.  $F$  may be any point of the ray other than  $E$ . For the moment, we may assume that  $F = G$ .

Now let  $I'$  denote the plane parallel to  $I$  and containing  $V$ . This plane divides the Euclidean three-space into two open half-spaces, one of which contains  $G$ . Let  $S$  denote this open half-space. Then for every point  $P$  of  $S$  there is a unique point  $P^*$  of  $I$  such that the ray from  $E$  through  $P$  meets  $I$  at  $P^*$ . The mapping,

$$S \rightarrow I; P \mapsto P^*$$

is the perspective transformation.

In NetLogo 3D, mouse operations and commands may be used to move the points  $V$  and  $G$ , while  $F$  is fixed at  $G$ . The default values are  $E = (0, 0, 10)$  and  $G = (0, 0, 0)$ .

## 5 References

NetLogo 3D 2.1pre1 README and User Manual (when and if)

Any text on linear algebra

Any text on computer graphics