The History of Path Separators

Why is the path separator a / on Unix/Linux/MacOS systems and \ on Microsoft Windows systems? The concept of a hierarchy of folders was first developed in Unix. On a Unix command line a / is used to separate folder names in a file path and dashes are used to specify command line options, e.g., path/to/file/myfile -long -reverse. On a Windows system the / character is used for command line options, so the designers of Windows decided to use the \ for separating folder names in a file path, e.g., path\to\file\myfile /long /reverse. Using a \ to separate folder names in a path is problematic because the \ character is also used as an escape character for special characters, such as \n for a new line character. Bottom line, we will always use the / character to separate folder names in a path, and even on Windows system the file path will work just fine.

222

This L-system looks very similar to the old L-system except that we’ve added one change. We’ve added the characters ‘[‘ and ‘]’. The meaning of these characters adds a very interesting new dimension to our L-Systems. The ‘[‘ character indicates that we want to save the state of our turtle, namely its position and its heading so that we can come back to this position later. The ‘]’ tells the turtle to warp to the most recently saved position. The way that we will accomplish this is to use lists. We can save the heading and position of the turtle as a list of 3 elements. [heading x y] The first index position in the list holds the heading, the second index position in the list holds the x coordinate, and the third index position holds the y coordinate.

Now, if we create an empty list and every time we see a ‘[‘ we append the list that contains [heading, x, y] we create a history of saved places the turtle has been where the most recently saved location will always be at the end of the list. When we find a ‘]’ in the string we use the pop function to remove the the most recently appended information.