11-21-2005
What did we talk about last class?
Since I'm going to be talking to Dr. Livingstone about BLAST tomorrow I figure we should hold off on doing that topic until then. Instead, we are going to do a fun overlap between Biology and CS today.
Since living without references is such a pain, let's talk really quickly about how to use references in Perl.

A reference is a scalar value that can refer to a scalar, a list, or a hash. You get a reference to a variable by prefixing it with \.

Inside of a function you treat the reference like a scalar, but when you want to pull out what it refers to you put an extra special symbol in front. You use $$ to dereference a scalar reference, @$ to dereference a list and %@$ for a hash.
L-Systems were developed by Lindenmayer to model the growth of various biological systems. They are grammars much like what we discussed before for Chomsky grammars. However, in an L-system, all the characters have productions applied to them in parallel. Some characters can become themselves.

There are numerous types of L-systems. They can be context free or context sensitive. They can be deterministic or stochastic. For more power they can be parameterized or extended.
Most applications of L-systems involve converting the strings over to some type of graphical representation. This is typically done with turtle graphics.

There are two characters for move forward and draw a line forward (F, f). There are also various characters for rotating either in 2-D or 3-D space (+, -, ...). Lastly, they typically have some type of push and pop characters for the positions of things ( [, ]).
This figure shows some simple grasses generated with L-systems.
Let's go look at some L-systems stuff now and write some code in Perl to do L-systems as strings. We don't have any ability to draw in Perl so we can't render those strings unless we write to file and then read it back in in a different language.
■ Remember that assignment #9 is due today.