COM 474 NATURAL COMPUTING TERM PROJECT REPORT

Artificial plant generation using D0L – Systems with Turtle Graphics

Hilal INAN & Feyza GALIP

Introduction

Our purpose for this project is to draw artificial plant with L-system grammar by using “jturtle” library in JAVA.

An L-system or Lindenmayer system is a parallel rewriting system and a type of formal grammar.

An L-system consists of an alphabet of symbols that can be used to make strings, a collection of production rules that expand each symbol into some larger string of symbols, an initial "axiom" string from which to begin construction, and a mechanism for translating the generated strings into geometric structures.

“jturtle” is a library which includes ready methods to plot lines to turtle.

Details

L-System

A model of morphogenesis, based on formal grammars (set of rules and symbols)

Introduced in 1968 by the Swedish biologist A. Lindenmayer

Originally designed as a formal description of the development of simple multi-cellular organisms

Later on, extended to describe higher plants and complex branching structures.

D0L-system

D0L-Systems is the simplest class of L-systems, termed D0L-systems (Deterministic and context free).

Lets us consider strings built of two letters a and b (they may occur many times in a string).

For each letter we specify are writing rule.

The rule a→ a b means that the letter a is to be replaced by the string ab, and the rule b→a means that the letter b is to be replaced by a. There writing process starts from a distinguished string called the axiom. Let us assume that it consist of a single letter b.

In the first derivation step (the first step of rewriting) the axiom b is replaced by a using production b→a. In the second step a is replaced by abusing the pro-duction a→ab. The word ab consist of two letters, both ofwhich are simultaneously replaced in the next derivation step. Thus, a is replaced by ab, b is replaced by a, and the string ab a results. In a similar way (by the simultaneous replacement of all letters), the string ab a yields abaab which in turn yields abaababa, then abaababaabaab, and so on.

Turtle Interprataion of the Strings

F Move forward a step of length *d*. The state of the turtle changes to *(x',y',α),* where x'= x + d cos(α) and y'= y + d sin(α). A line segment between points *(x,y)* and *(x',y')* is drawn

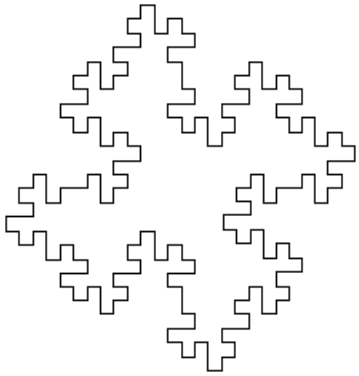
+ Turn right by angle *δ*. The next state of the turtle is *(x,y, α + δ)*

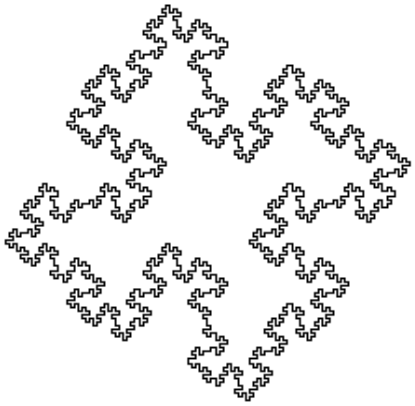
- Turn left by angle *δ*. The next state of the turtle is *(x, y, α -b)*

w: F+F+F+F

p: F →F+F-F-FF+F+F-F

Angle (*δ*) = 90º





Quadratic
Koch island

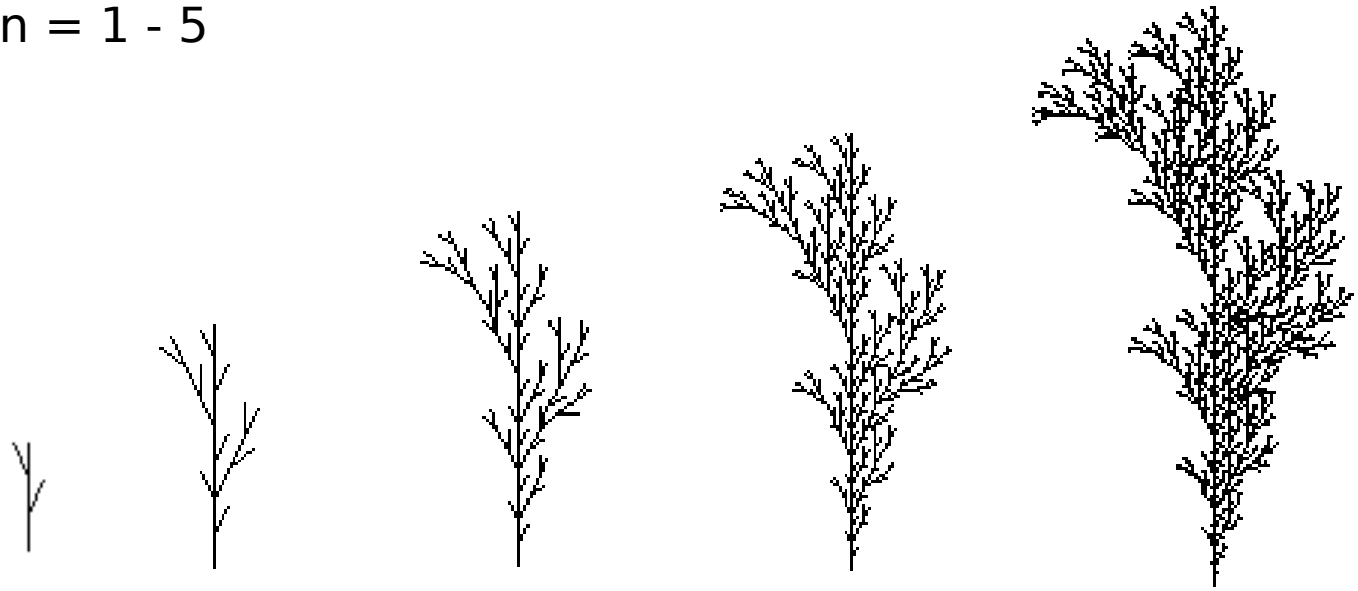
Turtle Interpretation of Bracketed Strings

w: F

p: F → F[-F]F[+F][F]

Angle (δ) = 60º

n = 1 - 5



jTurtle-0.1.1.jar LIBRARY

Some code examples from this library.
Turtle motion
Move and draw
forward() | fd()
backward() | bk() | back()
right() | rt()
left() | lt()
goto() | setpos()
Tell Turtle’s state

position() | pos()

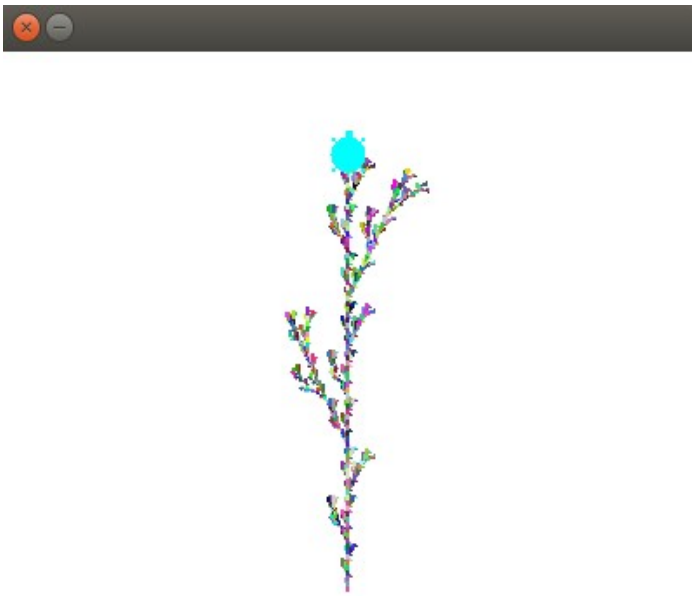
xcor()

ycor()

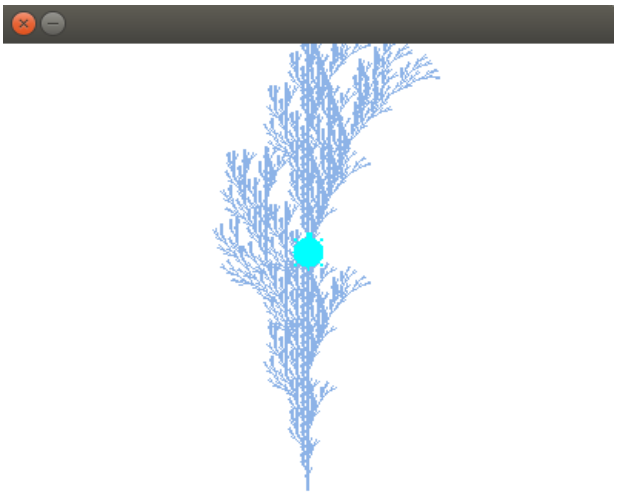
heading()

Setting and measurement [degrees()](#)
[radians()](#)

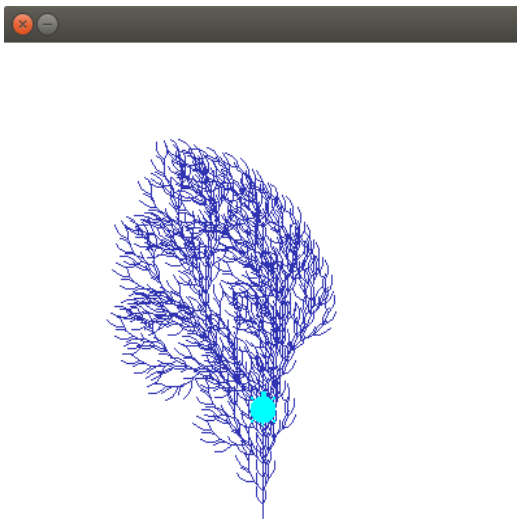
OUTPUTS FROM PROJECT



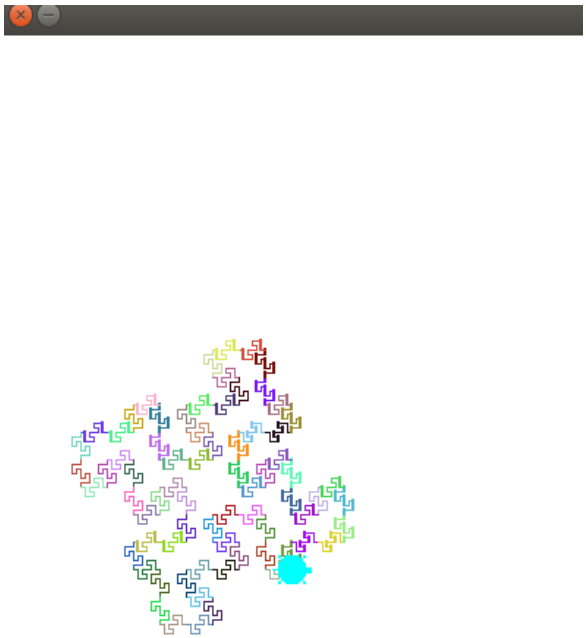
Angle = 25.7 n=5
F
F->F[+F]F[-F]F



Angle =20 n=5
F
F->F[+F]F[-F]F



Angle=22.5 n=4
F
F->FF-[-F+F+F]+[+F-F-F]



Angle = 90 n=2
F-F-F-F
F->F+FF-FF-F-F+F+FF-F-F+F+FF+FF-F

CONCLUSION

We have developed a program which ensures to draw artificial plants with “jturtle” library in JAVA by getting the L-system rule from user.

In addition the other shapes which are represented with L-system can be drawn.

References

1. http://en.wikipedia.org/wiki/L-system
2. http://ldc.usb.ve/~gabro/other/L-systems.ppt
3. http://www.java-online.ch/lego/legoEnglish/turtleGrafik.php?inhalt_links=turtle/nav_turtleTu.inc.php&inhalt_mitte=turtle/turtle.inc.php