Modélisation de la croissance d'une plante avec L-Py

OpenAlea

Modèles & softwares :

Core,

VisuAlea,

Standard Library,

SconsX,

Virtual Plants,

Alinea,

Mars-alt

…

Packages :

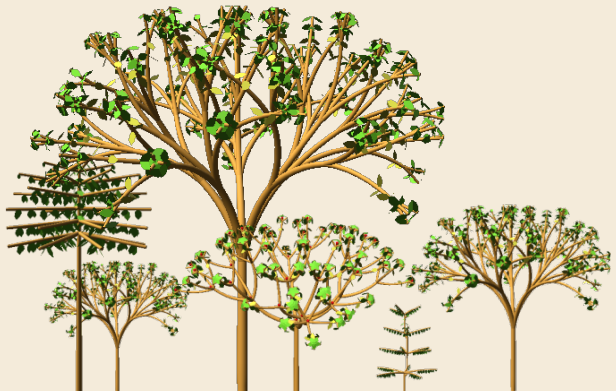
AML,

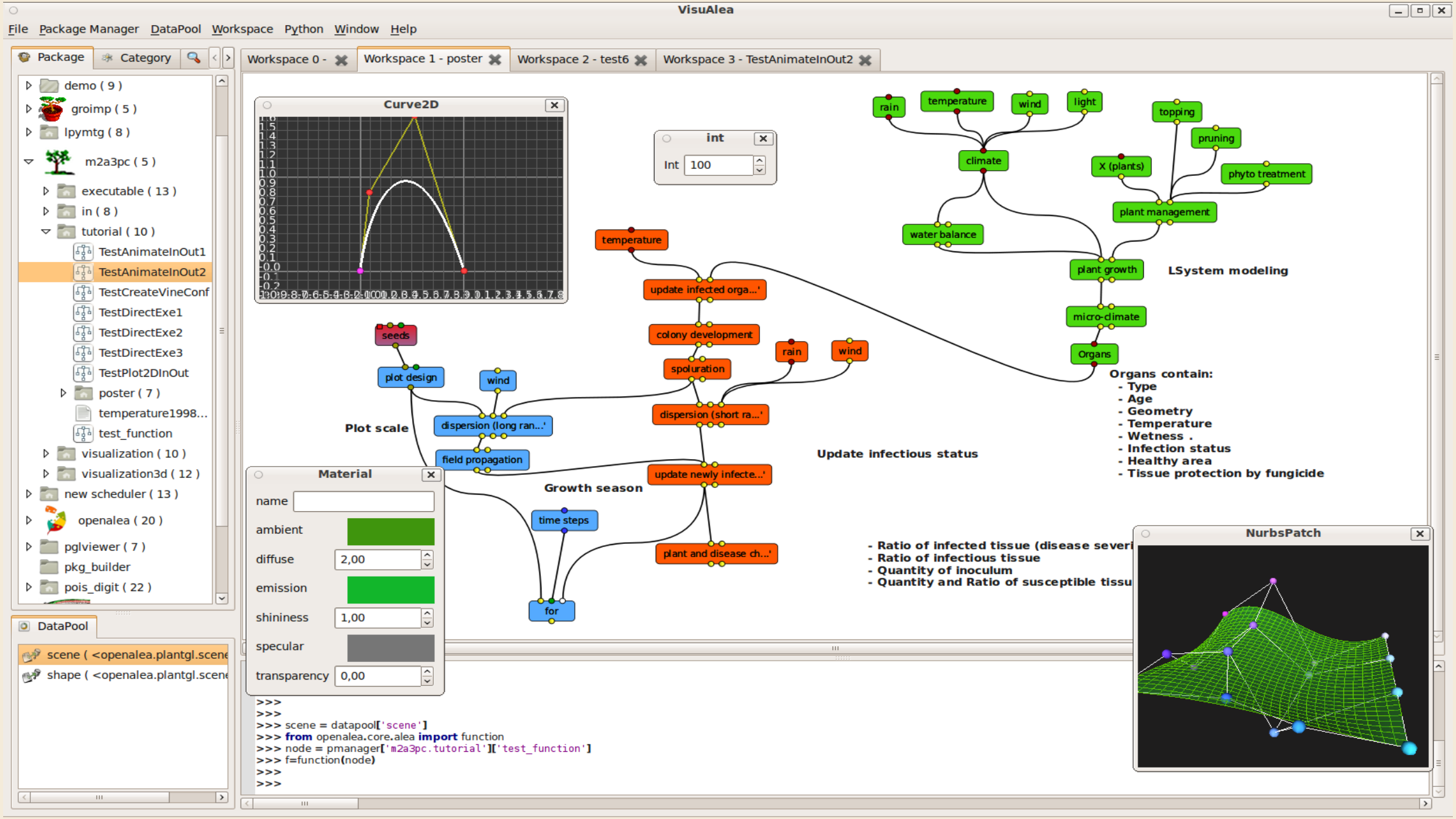
PlantGL,

L-py

…

…




VisuAlea
File Package Manager DataPool Workspace Python Window Help
Package
Category
demo (9)
groimp (5)
lpymtg (8)
m2a3pc (5)
executable (13)
in (8)
tutorial (10)
TestAnimateInOut1
TestAnimateInOut2
TestCreateVineConf
TestDirectExe1
TestDirectExe2
TestDirectExe3
TestPlot2DInOut
poster (7)
temperature1998...
test_function
visualization (10)
visualization3d (12)
new scheduler (13)
openalea (20)
pglviewer (7)
pkg_builder
pois_digit (22)
DataPool
scene (<openalea.plantgl.scene
shape (<openalea.plantgl.scene
Workspace 0 -
Workspace 1 - poster
Workspace 2 - test6
Workspace 3 - TestAnimateInOut2
Curve2D
int
Int 100
Material
name
ambient
diffuse 2,00
emission
shininess 1,00
specular
transparency 0,00
rain
temperature
wind
light
climate
topping
pruning
X (plants)
phyto treatment
plant management
water balance
plant growth
LSystem modeling
micro-climate
Organs
Organs contain:
- Type
- Age
- Geometry
- Temperature
- Wetness .
- Infection status
- Healthy area
- Tissue protection by fungicide
temperature
update infected orga...'
colony development
spoluration
rain
wind
dispersion (short ra...'
Update infectious status
update newly infecte...'
plant and disease ch...'
- Ratio of infected tissue (disease severi
- Ratio of infectious tissue
- Quantity of inoculum
- Quantity and Ratio of susceptible tissu
seeds
plot design
wind
Plot scale
dispersion (long ran...'
field propagation
Growth season
time steps
for
NurbsPatch
>>>
>>>
>>> scene = datapool['scene']
>>> from openalea.core.alea import function
>>> node = pmanager['m2a3pc.tutorial']['test_function']
>>> f=function(node)
>>>
>>>

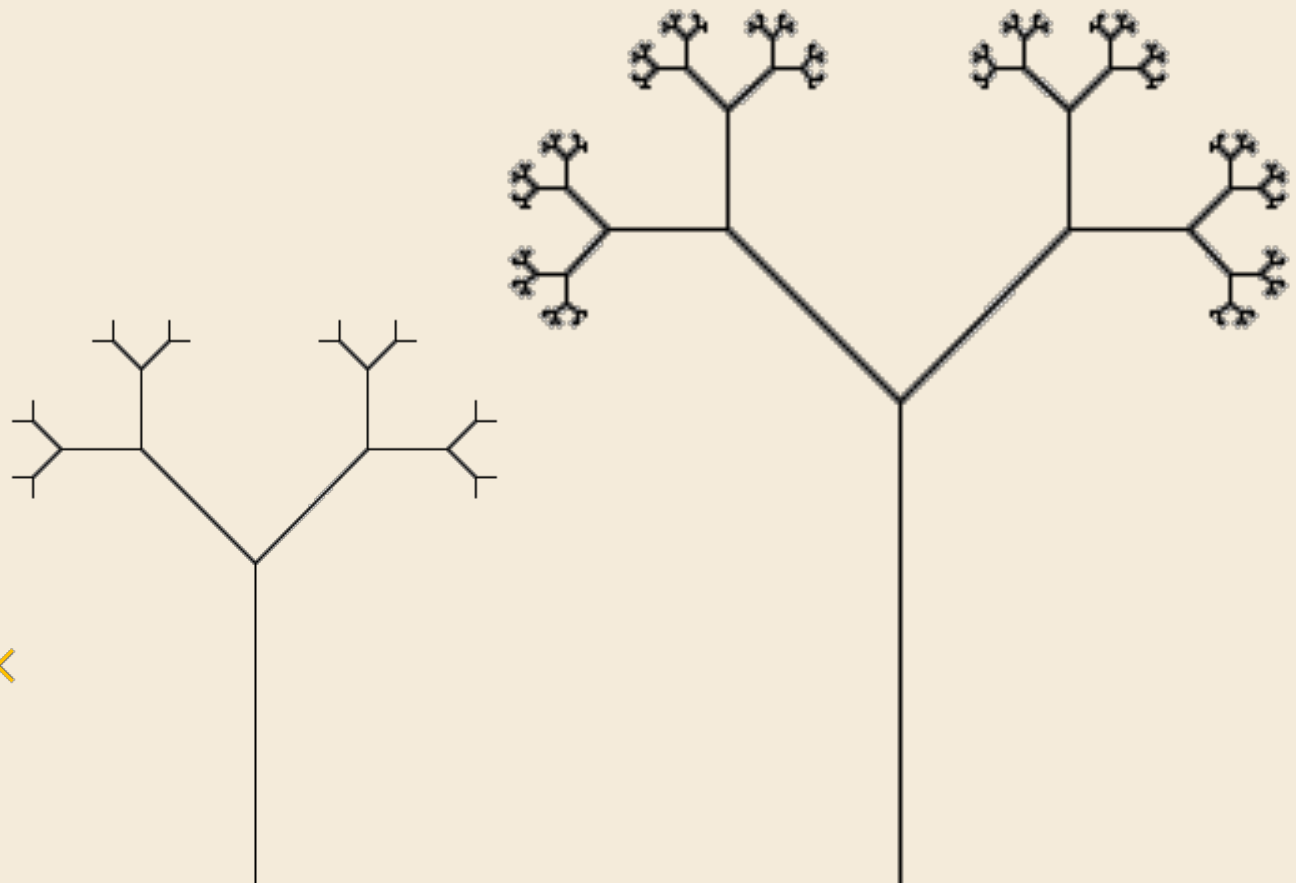
Système de Lindenmayer (L-system)

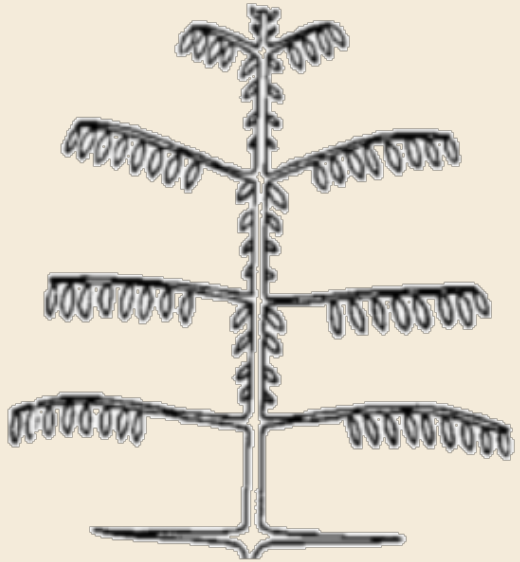
Règles: (A → AB), (B → A)

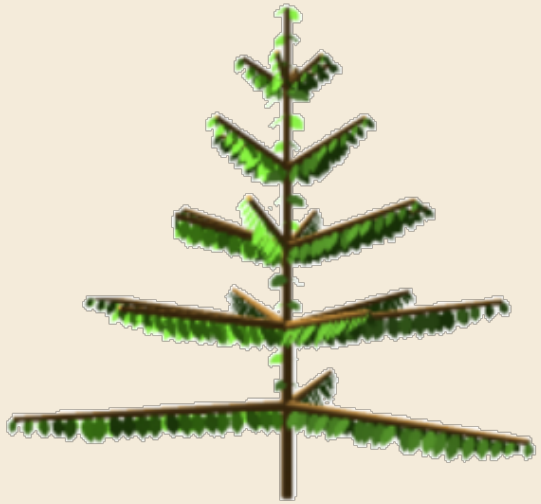
```
n=0:          A
             / \
n=1:        A   B
           /|    \
n=2:      A B     A
         /| |     |\
n=3:    A B A     A B
       /| | |\    |\ \
n=4: A B A A B    A B A
```

Règles : (1 → 11), (0 → 1[0]0)

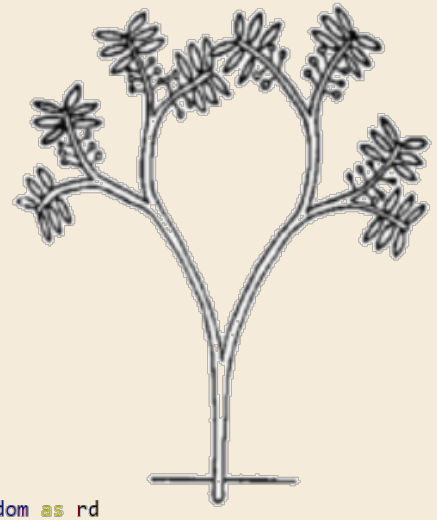
n=0: 0
n=1: 1[0]0
n=2: 11[1[0]0]1[0]0
n=3: 1111[11[1[0]0]1[0]0]11[1[0]0]1[0]0

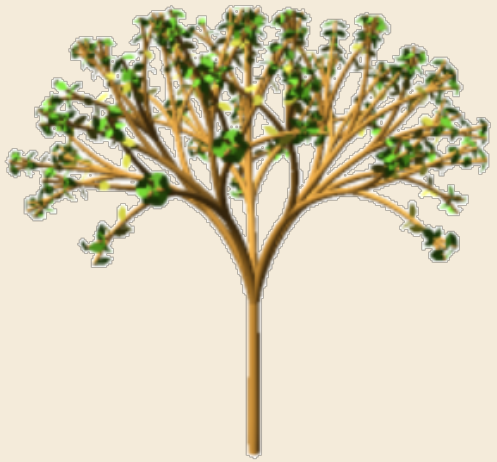






```
import random as rd

stagelength = 3
nbcycle = 8
leafduration = 20
leafmaturation = 5
angdev = 10
leafel = -60
radinc = 0.005

def maxleafsize(s,maxs):
  return ((maxs-s)*0.5/float(maxs))+0.5

def leafsize(maxsize, t):
  at = leafduration - t
  if at < leafmaturation:
    return maxsize * ((at*0.5/float(leafmaturation))+0.5)
  else:
      return maxsize

def branch_angle(nc):
    return 30+ 60*(nc/float(nbcycle))

Axiom: !(0.1)I(0.5,0.1)A(stagelength,nbcycle)

derivation length: nbcycle*stagelength
production:

A(t,nc) :
  if t == 0:
    if nc > 0:
     produce V(nc,rd.randint(3,6),0)A(stagelength,nc-1)
  else:
```

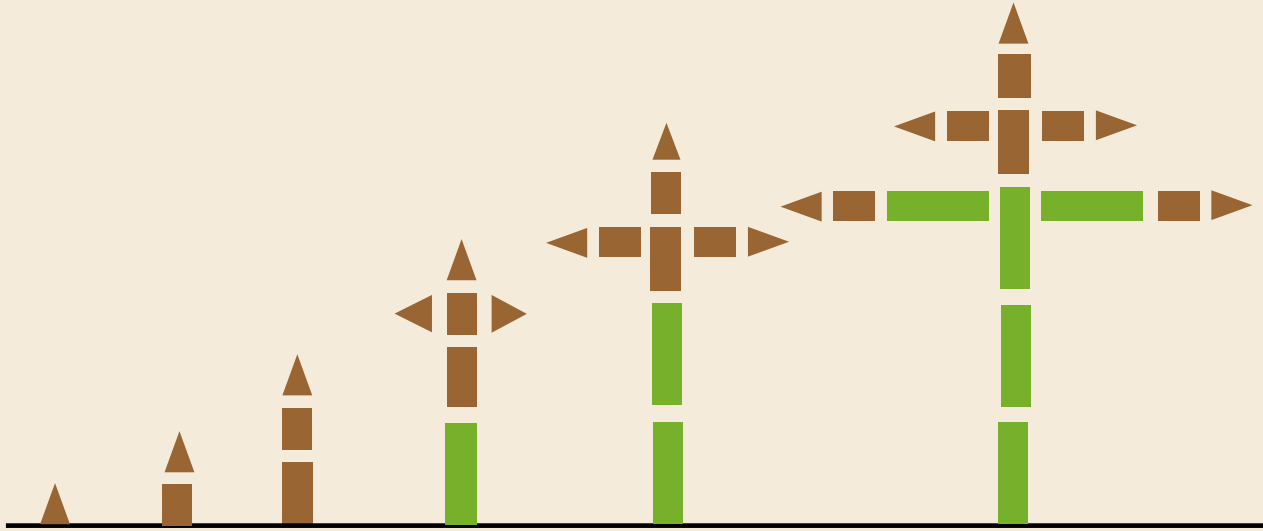


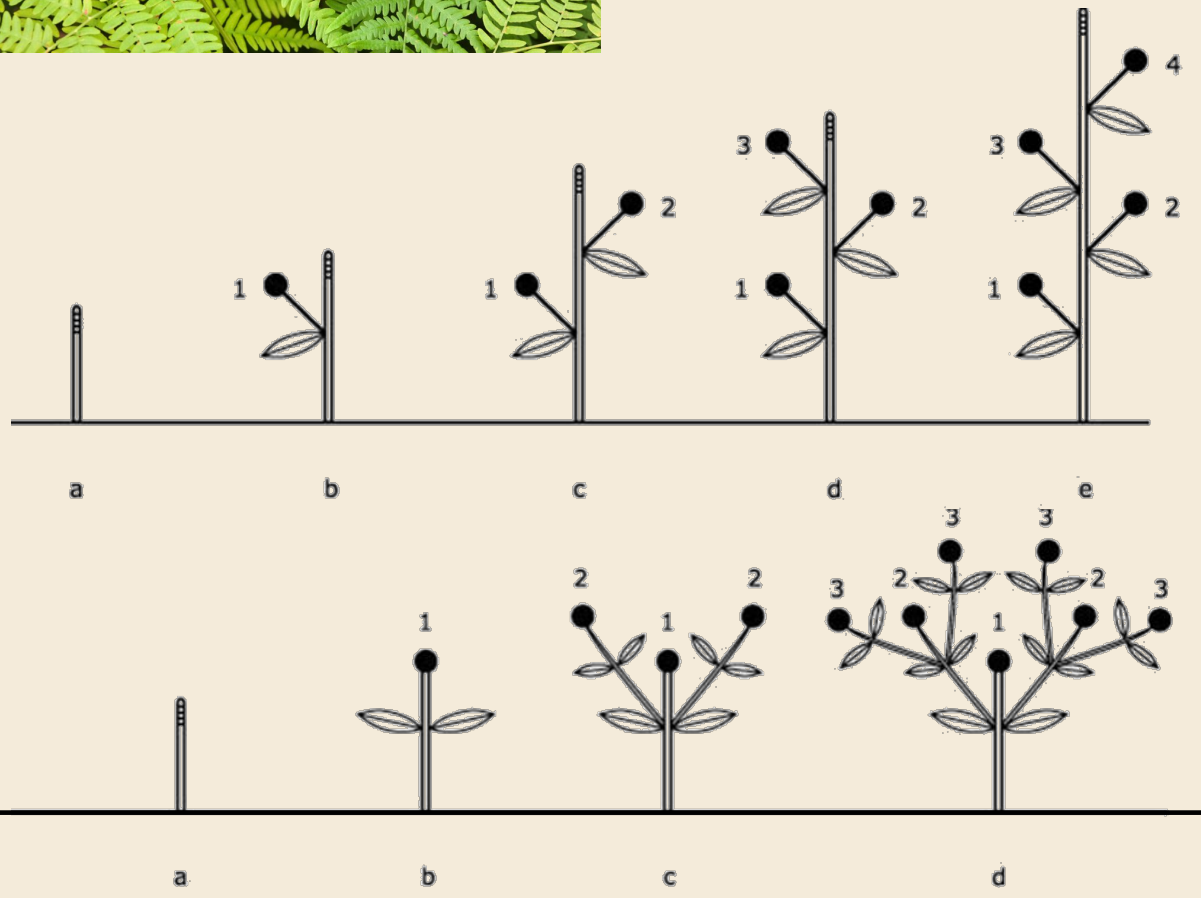


```
import random as rd

leafduration = 4 # life time of a leaf
leafold =      2 # age at which a leaf is considered as old
maxorder =     4 # maximum number of branching order
radinc =    0.01 # increment of radius through time

maxduration = lambda order : int(5./(order+1))+5  # life time of an apex
branch_angle = lambda order : 8 # branching angle
# number of axe at a ramification
nb_axes = lambda order : rd.randint(2,4) if order > 2 else rd.randint(4-int((order+1)/2),5-int((orde
up_angle = lambda t,order : -5        # up angle for lateral branches

# number total of iterations of the system
nbiter = sum([maxduration(o) for o in xrange(maxorder+1)])

module A # represent trunk apical meristem
module B # represent apical meristem of lateral branches
module L # whorl of leaf
module I # Internode


Axiom: _(0.05)@GcI(0.5,0.05)A(0,0)

derivation length: nbiter
production:

A(t,o) :
  if t < maxduration(o):
     # simply produces a metamer and ages the apex
```

Comme la Nature, basé sur l'itération



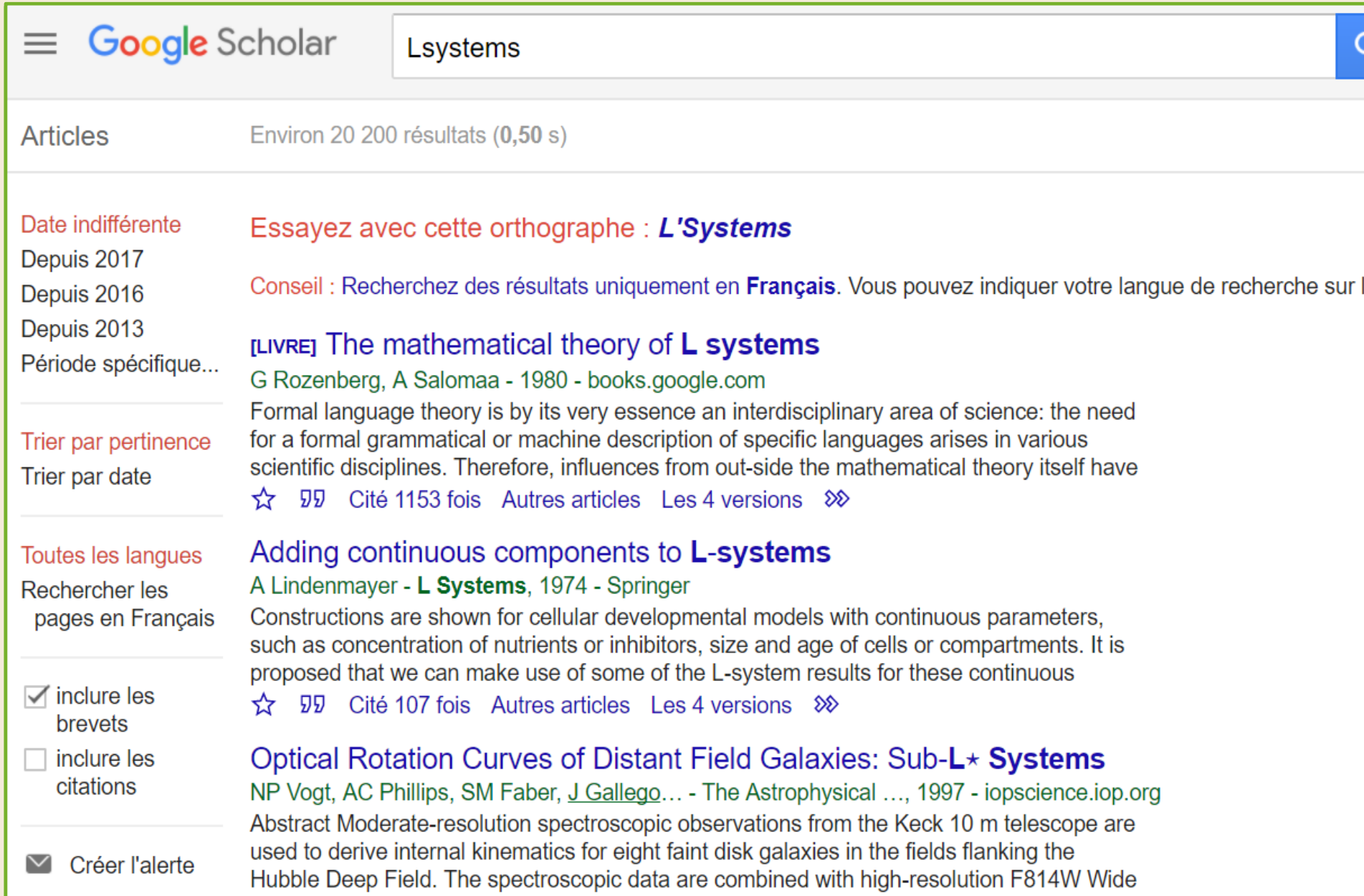





Comment ça marche ?

Dans la littérature

- Christophe Godin
- Christophe Pradal
- …



COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 132 (2016) 1–10

ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Developmental model of an automatic production of the human bronchial tree based on L-system

Amirabbas Davoodi, Ramin Bozorgmehry Boozarjomehry *

Chemical and Petroleum Engineering Department, Sharif University of Technology, Azadi Av., Tehran, Iran

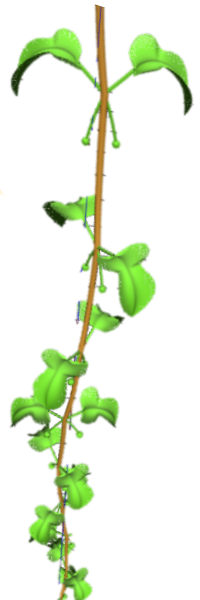
Notre Projet

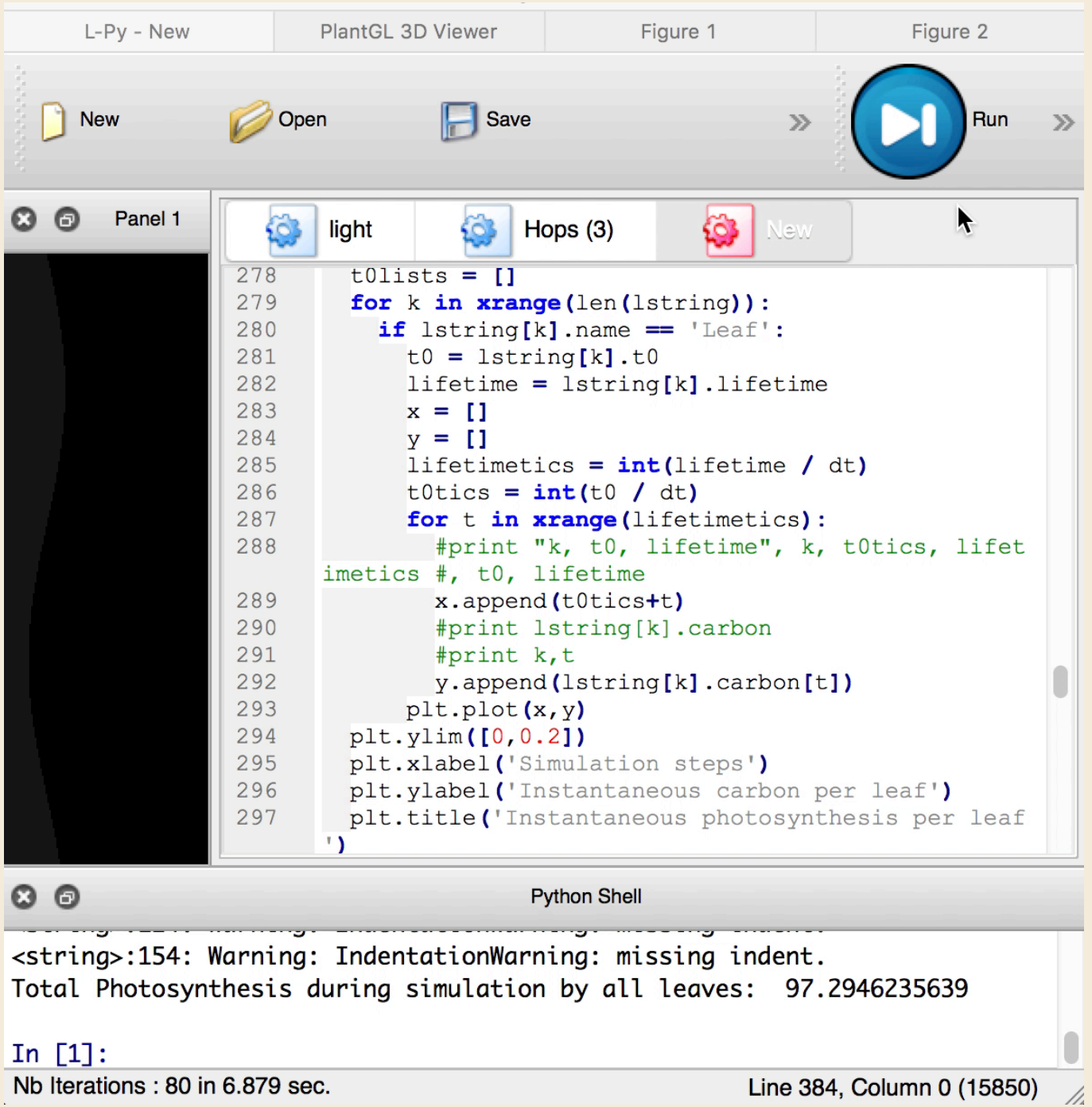
1ère étape : Comprendre L-Py et son langage ✓

2ème étape : Créer une plante modèle (croissance simulée) ✓

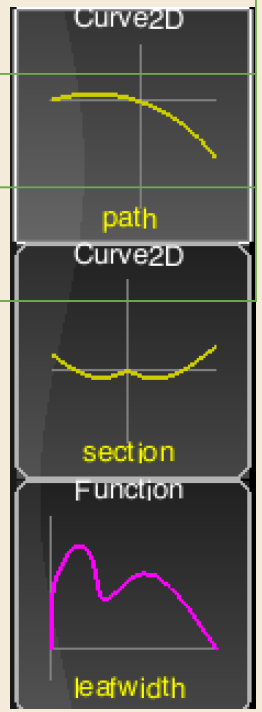
3ème étape : Implanter la plante dans le modèle de lumière ≈

4ème étape : Vérification avec données réelles ✗




L-Py - New
PlantGL 3D Viewer
Figure 1
Figure 2
New
Open
Save
Run
Panel 1
light
Hops (3)
New
t0lists = []
for k in xrange(len(lstring)):
if lstring[k].name == 'Leaf':
t0 = lstring[k].t0
lifetime = lstring[k].lifetime
x = []
y = []
lifetimetics = int(lifetime / dt)
t0tics = int(t0 / dt)
for t in xrange(lifetimetics):
#print "k, t0, lifetime", k, t0tics, lifetimetics #, t0, lifetime
x.append(t0tics+t)
#print lstring[k].carbon
#print k,t
y.append(lstring[k].carbon[t])
plt.plot(x,y)
plt.ylim([0,0.2])
plt.xlabel('Simulation steps')
plt.ylabel('Instantaneous carbon per leaf')
plt.title('Instantaneous photosynthesis per leaf')
Python Shell
<string>:154: Warning: IndentationWarning: missing indent.
Total Photosynthesis during simulation by all leaves: 97.2946235639
In [1]:
Nb Iterations : 80 in 6.879 sec.
Line 384, Column 0 (15850)

Problèmes rencontrés	Solutions trouvées
Représentation réaliste des feuilles	Utilisation de graphiques Polyline
Téléchargement de Open Alea	Utilisation d'un modèle fonctionnant sous L-Py
Fonctionnement du modèle de lumière sur PC	Utilisation uniquement sur Mac
Intégration de notre plante dans le modèle de lumière	Pas de solutions trouvées…



Conclusion

- Bugs d’installation, de codages,…
- Modèle ≠ réalité
- Importance de la communication
- Interaction de différents modèles