Capsis Training Exercices

fc - dec 2010

1. Capsis installation

- Install jdk1.6.x
- Add jdk1.6.x/bin at the beginning of the PATH
- Capsis complete local copy needed -> make a copy with a usb key
  - The local copy will not be synchronised with SVN
  - **The trainees must respect the Capsis charter and must not give the local copy to anyone (consider to remove it after a while)**
- Editor Textpad / Notepad++ (Windows) / Scite (Linux)
- Integrated compilation in the editor / ant in the terminal

2. Create a new module called 'Training' in Capsis

- ant createmodule -Dname=training -Dprefix=Tra -Dauthor=F._de_Coligny -Dinstitute=INRA
- Edit the etc/capsis.models file to add an entry for the new module
- Customize idcard.properties (Type + Description)
- ant compile
- Capsis > Help > About > select 'Training' in the list
- Test the module under Capsis, load the input file: capsis4/data/template/template.inv

3. Exercices on the 'Training ' module

a. Random regeneration

- Add a regeneration method to add new trees each year
- dbh : 3 cm
- height : 1.3 m
- Position: randomly on the terrain
- Only the trees greater than a given height: regenerationHeight (m) chosen at the beginning of the simulation may have children
- Number: randomly between 0 and a given number: regenerationMax, maximum children number per tree at each step, chosen by the user at the beginning of the simulation
- Other properties: free

Help:

- ✔ Add regenerationMax in the InitialParameters (mind translation, default value)
- ✔ Work in TraModel
- ✔ Scene dimension: xSize, ySize and origin (m)
- ✔ Use a random number generator java.util.Random to get the positions

- ✔ nextInt (max): an int between 0 and max - 1
- ✔ nextDouble (): a double between 0 and 1 (1 not included)
- ✔ Unique identifiers for the trees: treeIdDispenser.getNext ()
- ✔ Add a tree in the scene: TraScene.addTree (newTree)

b. N / Time Graph

- See the evolution of the number of trees during the simulation
- There exists a graph N / Time in Capsis, but it is not in the list of graphs compatible with 'Training '
- Adapt the TraMethodProvider to make this DETimeN graph compatible

Help:
- ✔ See MountMethodProvider (mountain.model)
- ✔ Use the NProvider interface (capsis.util.methodprovider)

c. Intervention

- Build an 'Intervener' extension to cut trees
- Cut the trees with a diameter below a given d diameter, with a given p probability

Help:
- ✔ Declare the thinning tool TraThinner in etc/extension.list (by analogy with C2Thinner)
- ✔ Check that it is found by Capsis in Tools > Extension Manager > Interveners (class name: TraThinner)
- ✔ Try it on a 'Training' project -> it cuts all the trees with a diameter below the chosen one
- ✔ Change TraThinner by analogy to add the probability

d. Add a terrain made of square cells

- After the loading of the initial scene
- Build a RectangularPlot made of SquareCells
- The cell width may be chosen at the beginning of the simulation
- Enlarge the scene size if needed to a multiple of the cell width
- Check the result in the 'Simple Viewer'

Help:
- ✔ Work in TraModel.initializeModel ()
- ✔ See TreeList.createPlot (GModel model, double cellWidth) (capsis.defaulttype)

e. Mortality

- Add a mortality method for the trees at each timestep
- You may sort the trees by size and use a probability

Help:

- ✔ Sort the trees on ascending dbh
- ✔ Collections.sort (List, Comparator) with capsis.util.GTreeDbhComparator
- ✔ Iterate on them and decide which ones should die
- ✔ Remove them from the list: for (Iterator i = trees.iterator (); i.hasNext ();) { ... if () ... i.remove () ... }
- ✔ Probability: if (random.nextDouble () < p) // true with a probability of p (p between 0 and 1)
- ✔ Refinement: tune the probability according to the tree size

f. Script

- Adapt and execute the SimpleScript script (training.myscripts)
- Change the script to rely on /template/template.inv
- Limit the evolution to 5 years
- The script saves the project in a file -> reopen it in Capsis interactive mode
- Change the script to run a thinning (see c.) after the 5 years, then run one more 5 years evolution
- Check the simulation in interactive mode with the N / Time graph (see b.)

Help:

- ✔ See the available help on the Capsis web site > Documentation > Script mode
- ✔ sh capsis.sh -p script training.myscripts.SimpleScript

g. The trees under the crown of a neighbour grows twice slower

- Change the growth of the trees according to the local competition

Help:

- ✔ Add a maxCrownRadius (m) property to the scene, update it at each step
- ✔ For each tree, check the neighbours in this radius and consider those which crownRadius is greater than the distance from the tree to the neighbour
- ✔ Find the neighbours within a given radius with capsis.defaulttype.RoundMask and getTreesNear (t) (see comments inside RoundMask)
- ✔ Get the distance from a tree to one neighbour with RoundMask.getDistance (neighbour)
- ✔ Add a 'covered' property in the tree and consider it for its growth

h. DispersionRégénération par dispersion

- Add another regeneration function
- position: the new trees appear in a radius = 2 * crownRadius arround the parent (x0, y0)
- Choose the regeneration method at the beginning of the simulation

Help:

- ✔ Calculate the coordinates (x, y) of each new tree by drawing:
- ✔ a distance in the [0, maxDistance] interval
- ✔ an azimuth in the [0, 2PI[radians interval

- ✔ x = x0 + Math.cos (azimuth) * maxDistance
- ✔ y = y0 + Math.sin (azimuth) * maxDistance
- ✔ Add a 'dispersionRegeneration' boolean in the IntialParameters (mind traductions and default value)