Capsis Training Exercices

December 2017

François de Coligny - Nicolas Beudez

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

- in a terminal, adapt and type this command...

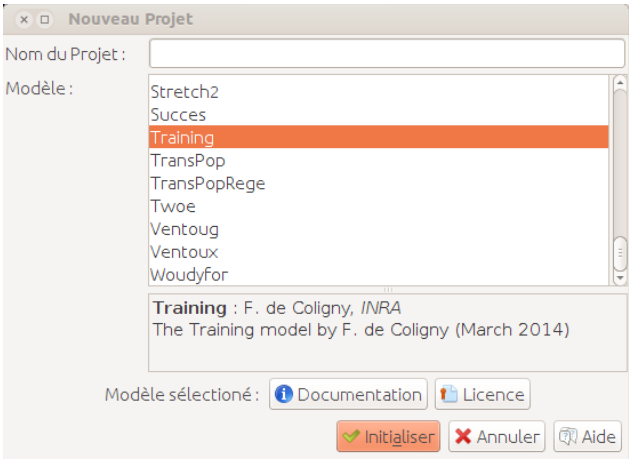
  For Windows :
  ant createmodule -Dname=training -Dprefix=Tra -Dauthor=F._de_Coligny -Dinstitute=INRA

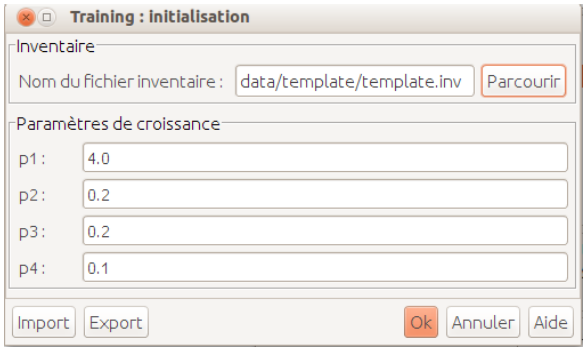
  For Linux/MacOS:
  sh ant createmodule -Dname=training -Dprefix=Tra -Dauthor=F._de_Coligny -Dinstitute=INRA

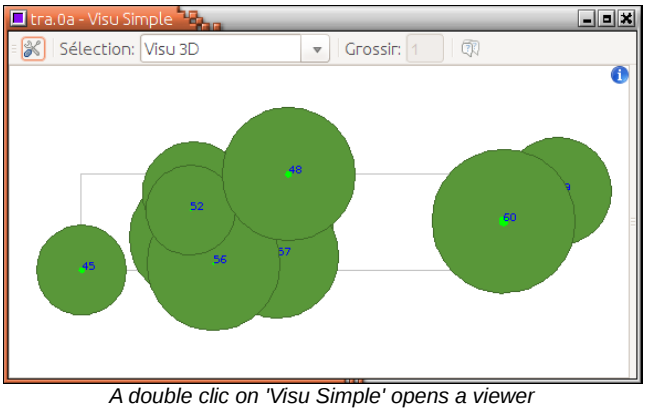
- you can verify that the etc/capsis.models file contains an entry for the new module (to do it, simply open this file)
- customize idcard.properties (type + description)
- compile...

  ant compile

- Capsis > Help > About > select 'Training' in the list
- test the module under Capsis...
  - load the input file: capsis4/data/template/template.inv
  - load a file created during the Java exercises session: trees.txt







A double clic on 'Visu Simple' opens a viewer

2. Random regeneration

Add a regeneration method to add new trees each year, located randomly on the terrain.

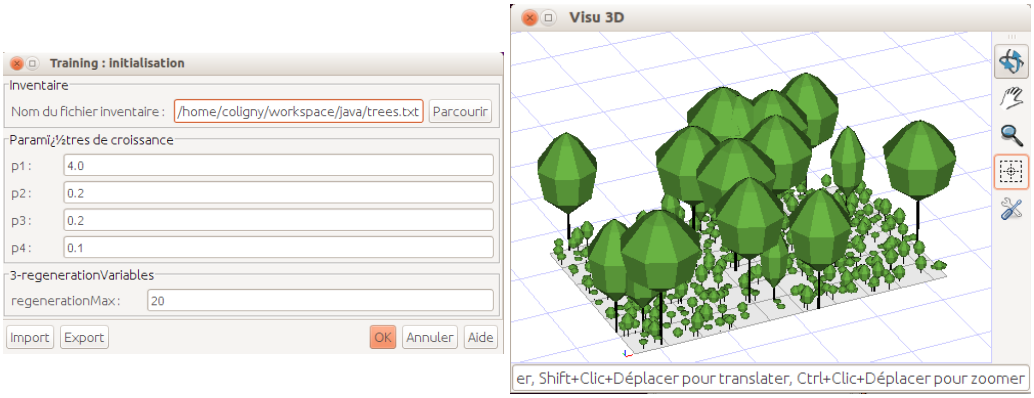
Notes:

- new trees dbh: 3 cm and height: 1.3 m
- how many: draw each year a random number between 0 and regenerationMax, chosen by the user at the beginning of the simulation
- choose new unique ids for the new trees

Helper:

- add regenerationMax in TraInitialParameters
- add a regeneration method in TraModel
- draw random numbers with java.util.Random
- make a loop to create and add new trees
- you may pick ideas below...

```
import java.util.Random;
import jeeb.lib.util.Vertex3d;

@Editor (group = "3-regenerationVariables")
public int regenerationMax;

private Random random;
random = new Random ();

// Call the regeneration method
regeneration (newScene);

int n = getSettings ().regenerationMax;
Vertex3d origin = newScene.getOrigin (); // m
double xSize = newScene.getXSize (); // m
double ySize = newScene.getYSize (); // m
int id = treeIdDispenser.next ();
double x = origin.x + random.nextDouble () * xSize;
TraTree t = new TraTree (id, newScene, age, height, dbh,
        crownBaseHeight, crownRadius, x, y, z);
newScene.addTree (t);
```



After 20 years

Add translations for the initial dialog

```
# In training/TraLabels_en.properties
3-regenerationVariables = Regeneration variables
regenerationMax = Max number of new trees each year

# In training/TraLabels_fr.properties
3-regenerationVariables = Régénération variables
regenerationMax = Nombre max de nouveaux arbres chaque année
```

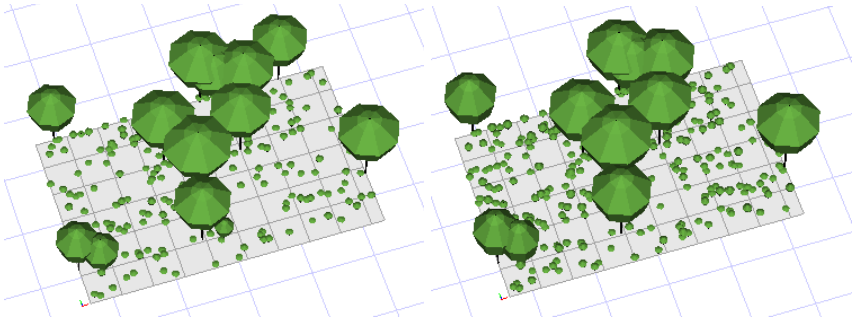
3. Mortality

Add some code in the Training module to remove trees, according to a global survival probability of 98 % each year.

Helper:

- trees that are not added in newScene 'die'
- you may pick ideas below...

```
double survivalProba = 0.98;
double proba = random.nextDouble (); // [0,1[
if (proba > survivalProba) ... // this tree is dead
```



Year 10 *Year 15*

4. Add a geometrical plot made of square cells

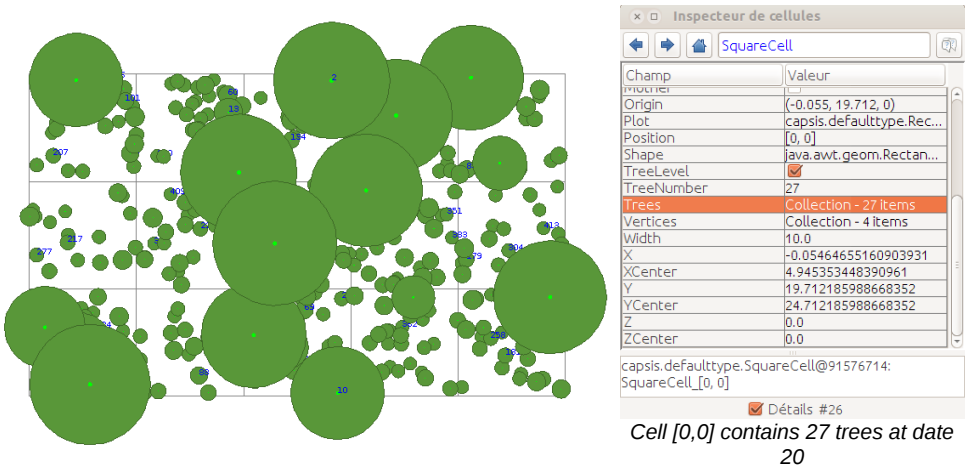
Use the default proposals of Capsis to build a rectangular plot made of square cells on the ground.

Helper:

- the rectangular plot must be constructed at the end of the project initialisation
- you may have a look in these classes to check the initialisation process...

```
TraInitialParameters
TraModel
TraInventoryLoader
```

- expected result (here 15 cells)



Cell [0,0] contains 27 trees at date 20

5. Make a graph: N / Time

Write adaptations to make the Capsis graph 'N / Time' compatible with the Training module.

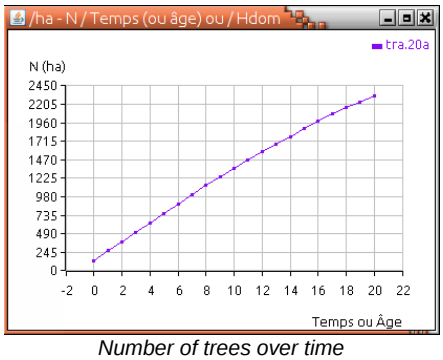
Notes:

- see capsis.extension.dataextractor.DETimeN
- the matchWith () method tests compatibility
- the referent is a reference to the model class (here TraModel)
- what do you have to change to make this chart compatible ?

Helper:

- do not change anything in the chart class (generic)
- the changes must be done in the Training module classes
- look at the TraMethodProvider...

```
capsis.util.methodprovider.NProvider;
public class TraMethodProvider implements ...NProvider... {

/**
 * Number of trees.
 */
public double getN (GScene stand, Collection trees) {
    if (trees == null) {return -1;}
    return trees.size ();
}
```



Number of trees over time

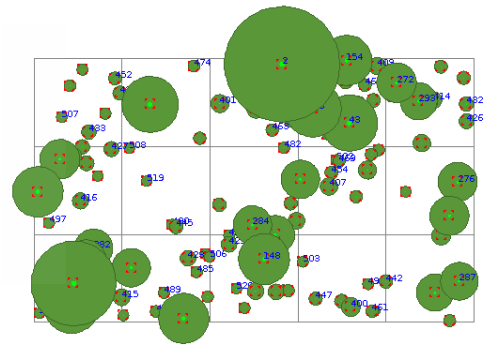
6. Script

Adapt the script example in training/myscripts/ to load your own treeFile, set the regenerationMax parameter to 5 per year, and change the evolution stage to reach 100 years, performing an intervention every 25 years.

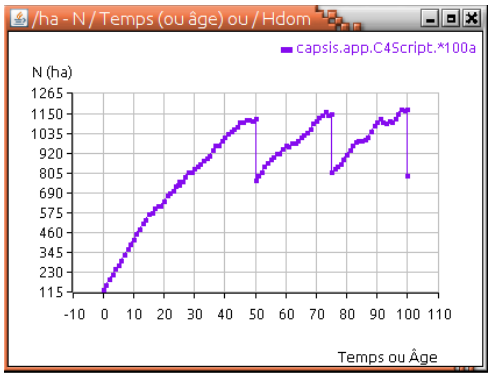
Helper:

- you may pick ideas below...

```
Linux: sh capsis.sh -p script training.myscripts.SimpleScript
Windows: capsis -p script training.myscripts.SimpleScript

import capsis.kernel.extensiontype.*;
import capsis.extension.intervener.*;

i.regenerationMax = 5;
s.init (i);

Step step = s.evolve (new TraEvolutionParameters (25));

// Intervention: cut tree between 5 and 15m height
Intervener thinner = new DHAThinner (DHAThinner.HEIGHT, 5, 15);
step = s.runIntervener (thinner, step);

date = step.getScene ().getDate ();
```



The final scene at year 100 after the last cut



Number of trees over time

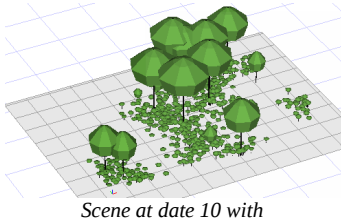
7. Regeneration around the mothers

Write an alternative regeneration method : only trees older than 20 years can regenerate, they give birth to at most regenerationMax trees and these new trees are located around their mother, at a max distance of n meters with n = mother height / 2.

Helper:

- you may pick ideas below...

```
List copy = new ArrayList (newScene.getTrees ());
for (Object o : copy) {
    TraTree mother = (TraTree) o;

double maxDistance = mother.getHeight () / 2d;
double x0 = mother.getX ();

for (int i = 0; i < n; i++) {
double distance = random.nextDouble () * maxDistance;
double alpha = random.nextDouble () * 2 * Math.PI;
double x = x0 + Math.cos (alpha) * distance;
newScene.addTree (t) ;...
```



Scene at date 10 with regenerationMax = 10