# 2016

## 01.02.2016 Préparation expertise JMD/LG Simulation proba montée = 20.000

<?xml version=*"1.0"* encoding=*"UTF-8"* ?>

<!-- fileName : parameters.xml -->

<!-- protocolName : CENTENAL -->

<parameters>

<!-- Conditions de simulation -->

<parameter name=*"PROTOCOL"* displayName=*"00/ PROTOCOL"* type=*"java.lang.String"*

defaultValue=*"CENTENAL"* isReadOnly=*"true"*

converter=*"repast.simphony.parameter.StringConverterFactory$StringStringConverter"* />

<parameter name=*"VERBOSE"* displayName=*"01/ Display events on console"*

type=*"boolean"* defaultValue=*"false"* isReadOnly=*"false"*

converter=*"repast.simphony.parameter.StringConverterFactory$BooleanConverter"* />

<parameter name=*"DISPLAY\_MAP"* displayName=*"01/ Display map over grounds"*

type=*"boolean"* defaultValue=*"true"* isReadOnly=*"false"*

converter=*"repast.simphony.parameter.StringConverterFactory$BooleanConverter"* />

<parameter name=*"TERMINATE"* displayName=*"12/ Terminate simulation"*

type=*"boolean"* defaultValue=*"false"* isReadOnly=*"false"*

converter=*"repast.simphony.parameter.StringConverterFactory$BooleanConverter"* />

<parameter name=*"RASTER\_FILE"* displayName=*"02/ Grid filename"*

type=*"java.lang.String"* defaultValue=*"20140527-centenalRaster.txt"*

isReadOnly=*"true"*

converter=*"repast.simphony.parameter.StringConverterFactory$StringStringConverter"* />

<!-- Soil cell parameters -->

<parameter name=*"CELL\_WIDTH\_Ucm"* displayName=*"03/ Centimeters in one pixel"*

type=*"java.lang.Integer"* defaultValue=*"750000"* isReadOnly=*"true"*

converter=*"repast.simphony.parameter.StringConverterFactory$IntConverter"* />

<parameter name=*"TICK\_LENGTH\_Ucalendar"* displayName=*"05/ Tick length (unit above)"*

type=*"java.lang.Integer"* defaultValue=*"11"* isReadOnly=*"false"*

converter=*"repast.simphony.parameter.StringConverterFactory$IntConverter"* />

<parameter name=*"TICK\_UNIT\_Ucalendar"* displayName=*"04/ Tick unit (s/m/h/d/mon/y)"*

type=*"java.lang.String"* defaultValue=*"d"* isReadOnly=*"false"*

converter=*"repast.simphony.parameter.StringConverterFactory$StringStringConverter"* />

<parameter name=*"VEHICLE\_LOADING\_PROBA\_DIVIDER"* displayName=*"09/ p(rodentClimb) (1-data,100-data/100,...)"*

type=*"java.lang.Integer"* defaultValue=*"10000"* isReadOnly=*"false"*

converter=*"repast.simphony.parameter.StringConverterFactory$IntConverter"* />

<parameter name=*"RODENT\_SUPER\_AGENT\_SIZE"* type=*"java.lang.Integer"*

defaultValue=*"1"* isReadOnly=*"false"* displayName=*"10/ Rodents super-agent size"*

converter=*"repast.simphony.parameter.StringConverterFactory$IntConverter"* />

<parameter name=*"HUMAN\_SUPER\_AGENT\_SIZE"* type=*"java.lang.Integer"*

defaultValue=*"10"* isReadOnly=*"false"* displayName=*"11/ carriers super-agent size"*

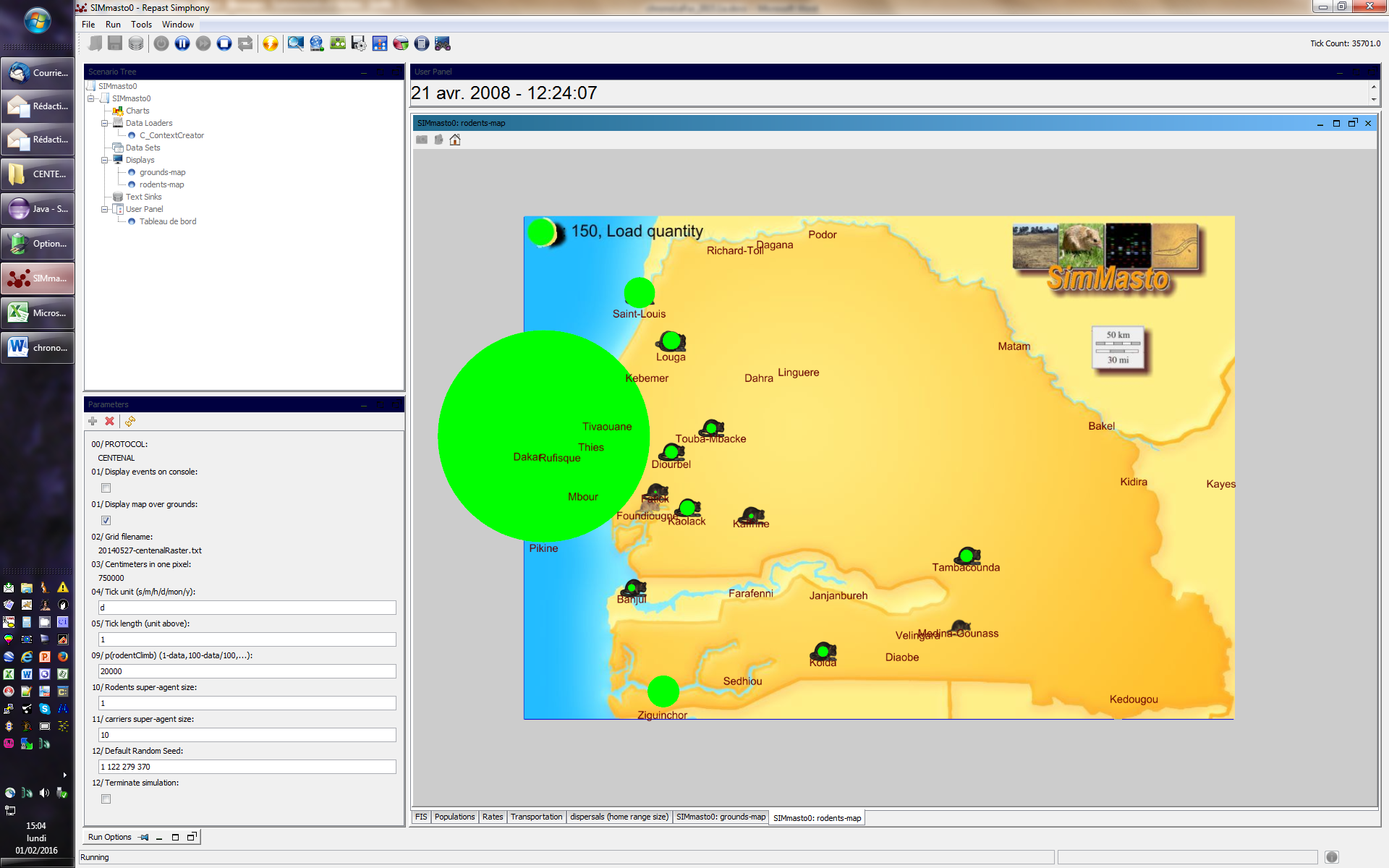
converter=*"repast.simphony.parameter.StringConverterFactory$IntConverter"* />

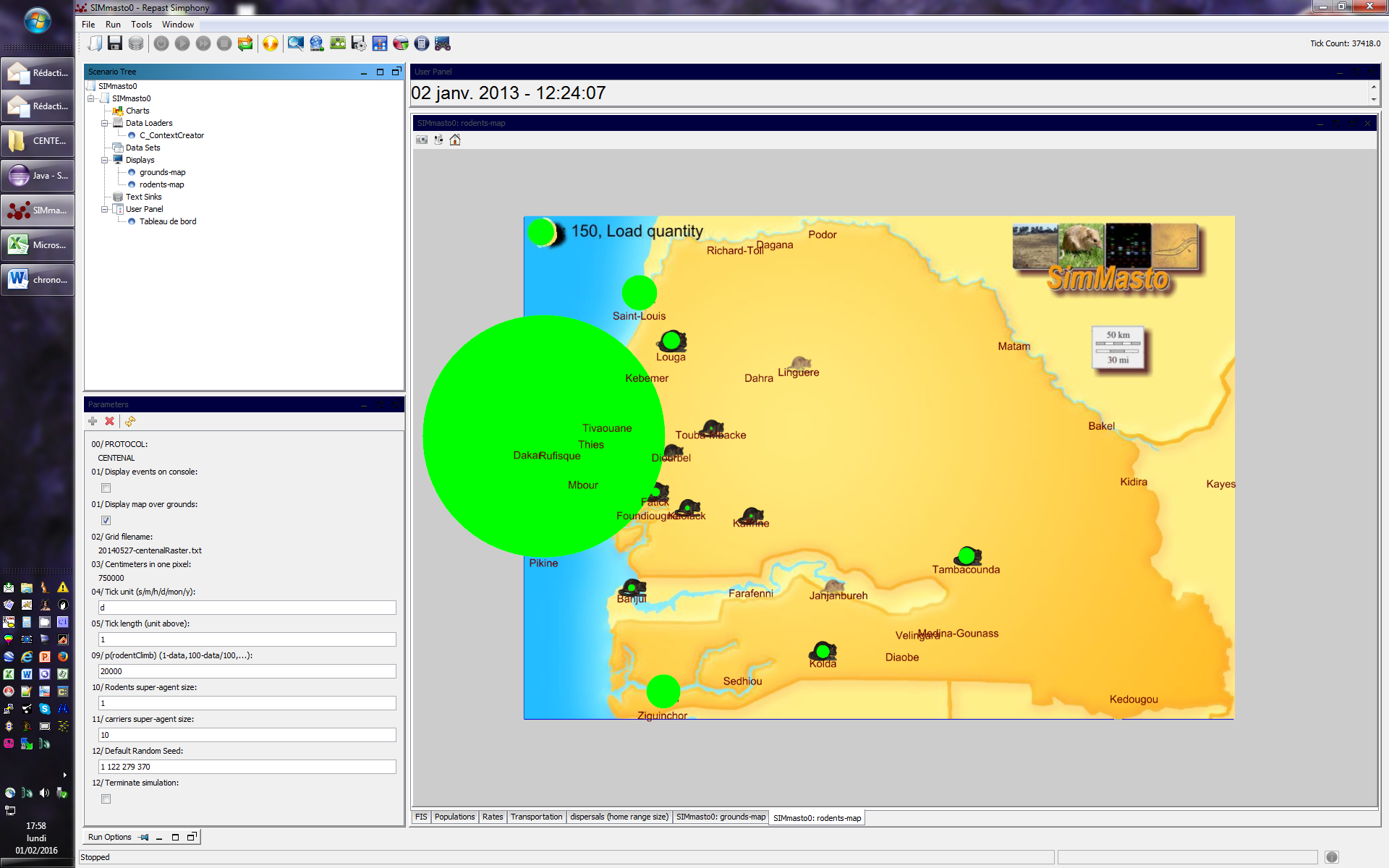
<parameter name=*"randomSeed"* displayName=*"12/ Default Random Seed"*

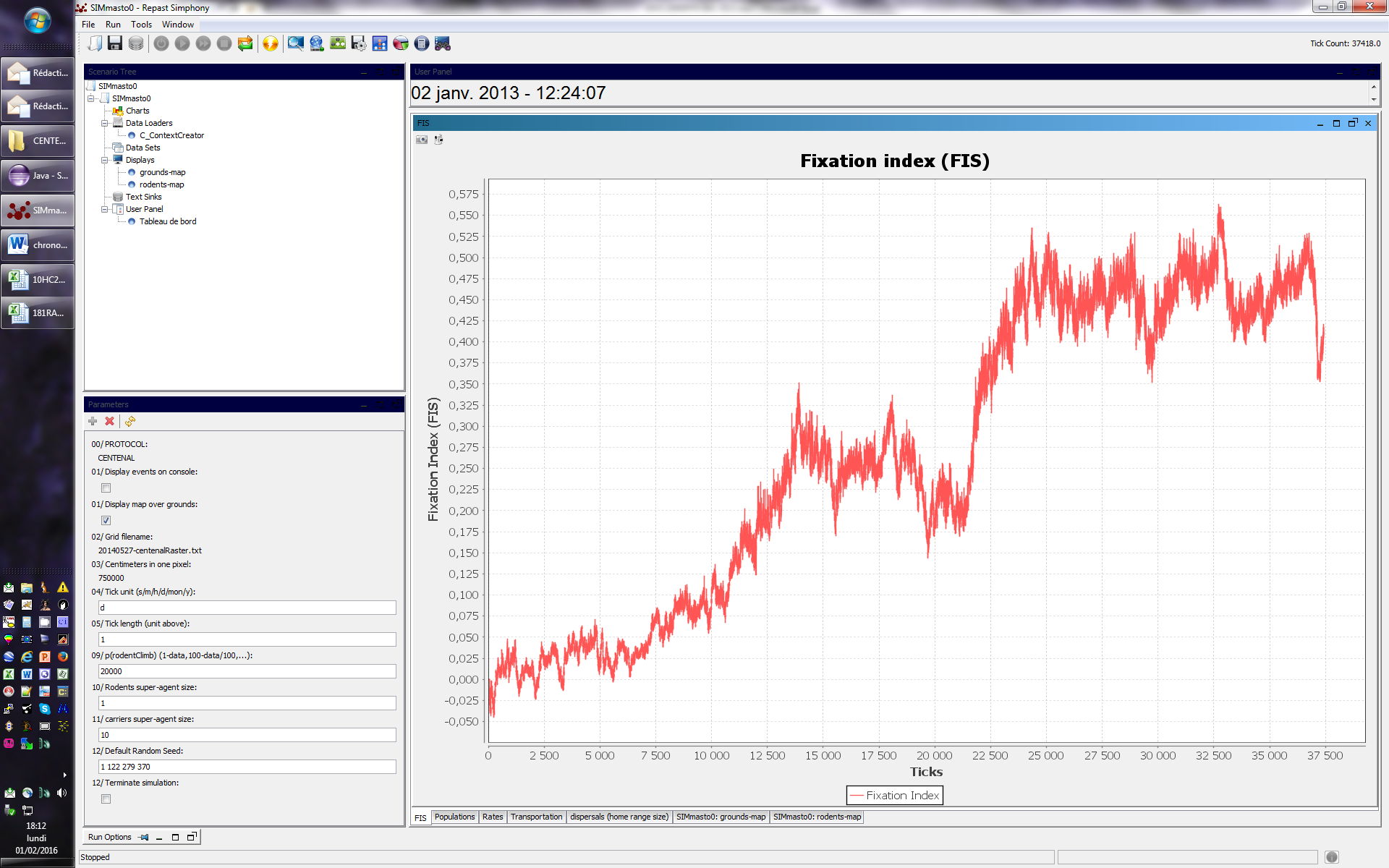
type=*"int"* defaultValue=*"1122279370"* isReadOnly=*"false"*

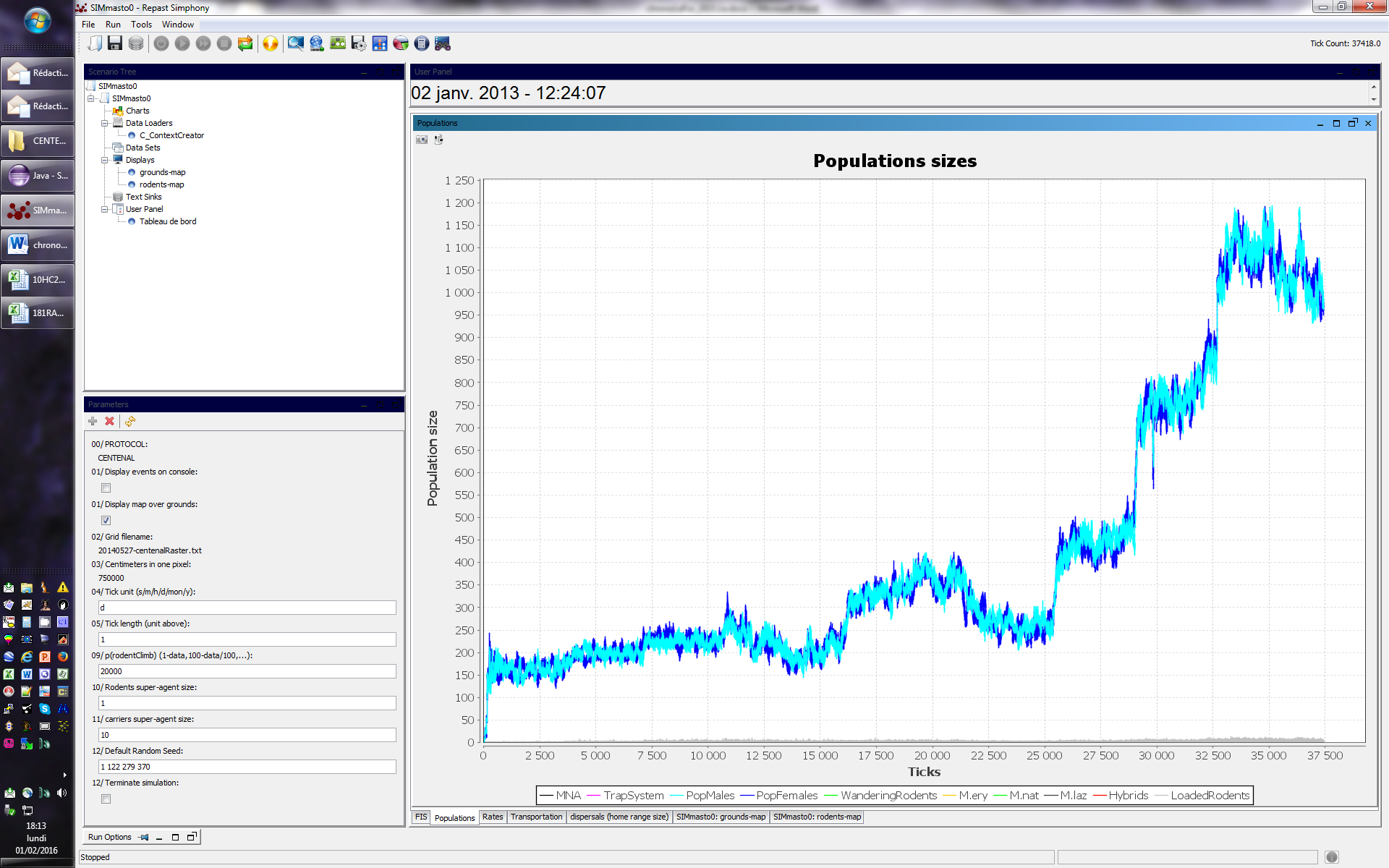
converter=*"repast.simphony.parameter.StringConverterFactory$IntConverter"* />

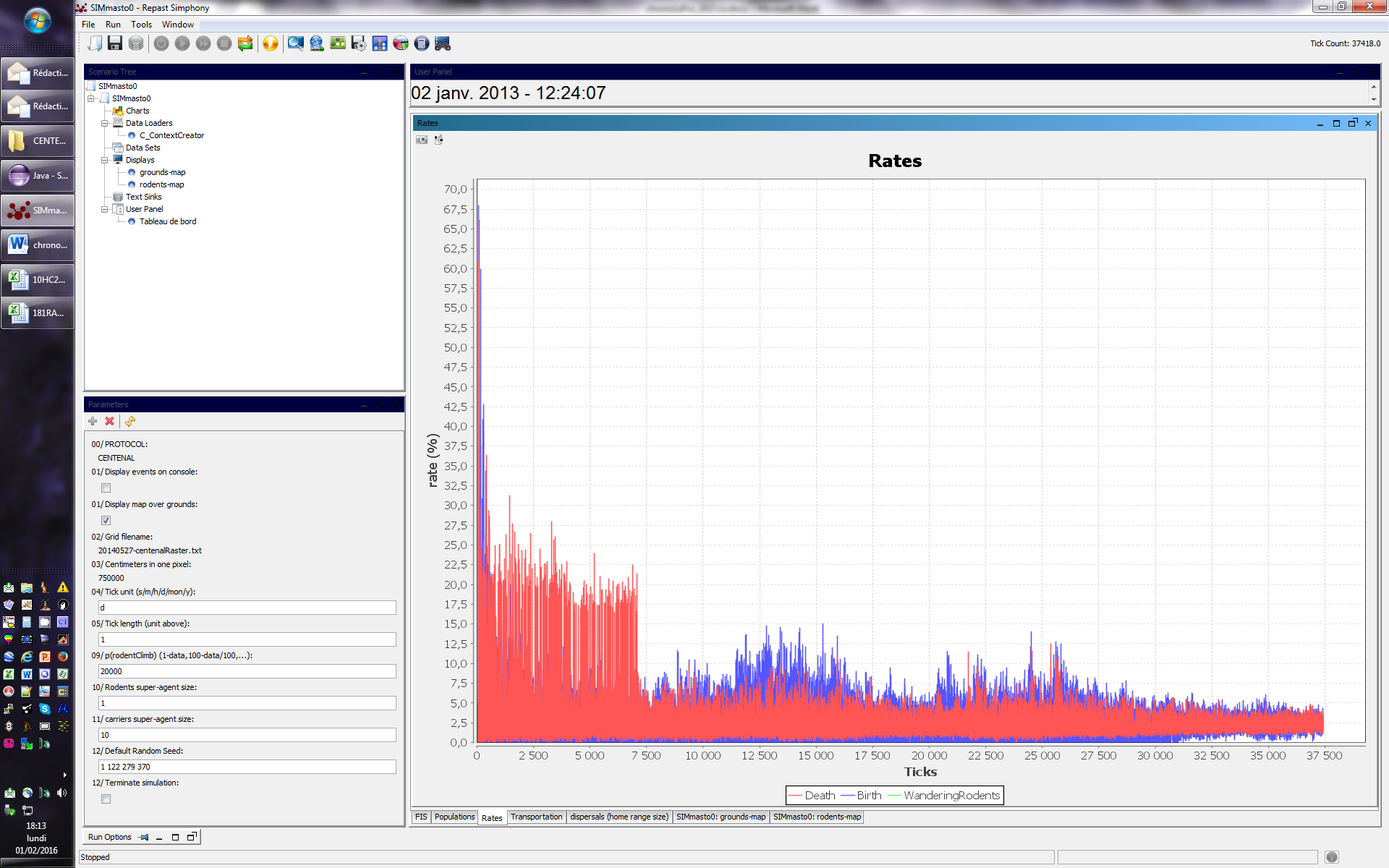
</parameters>

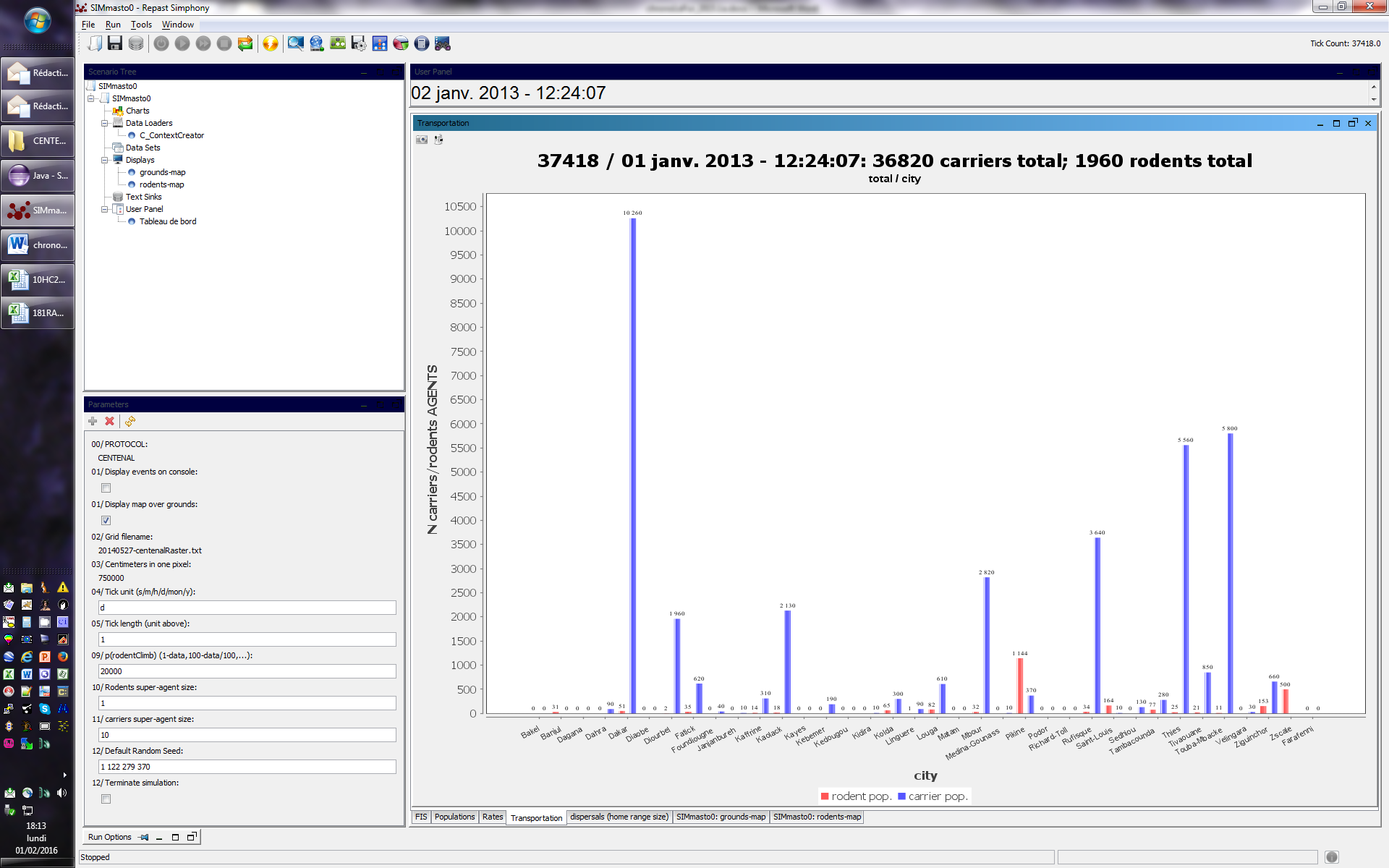


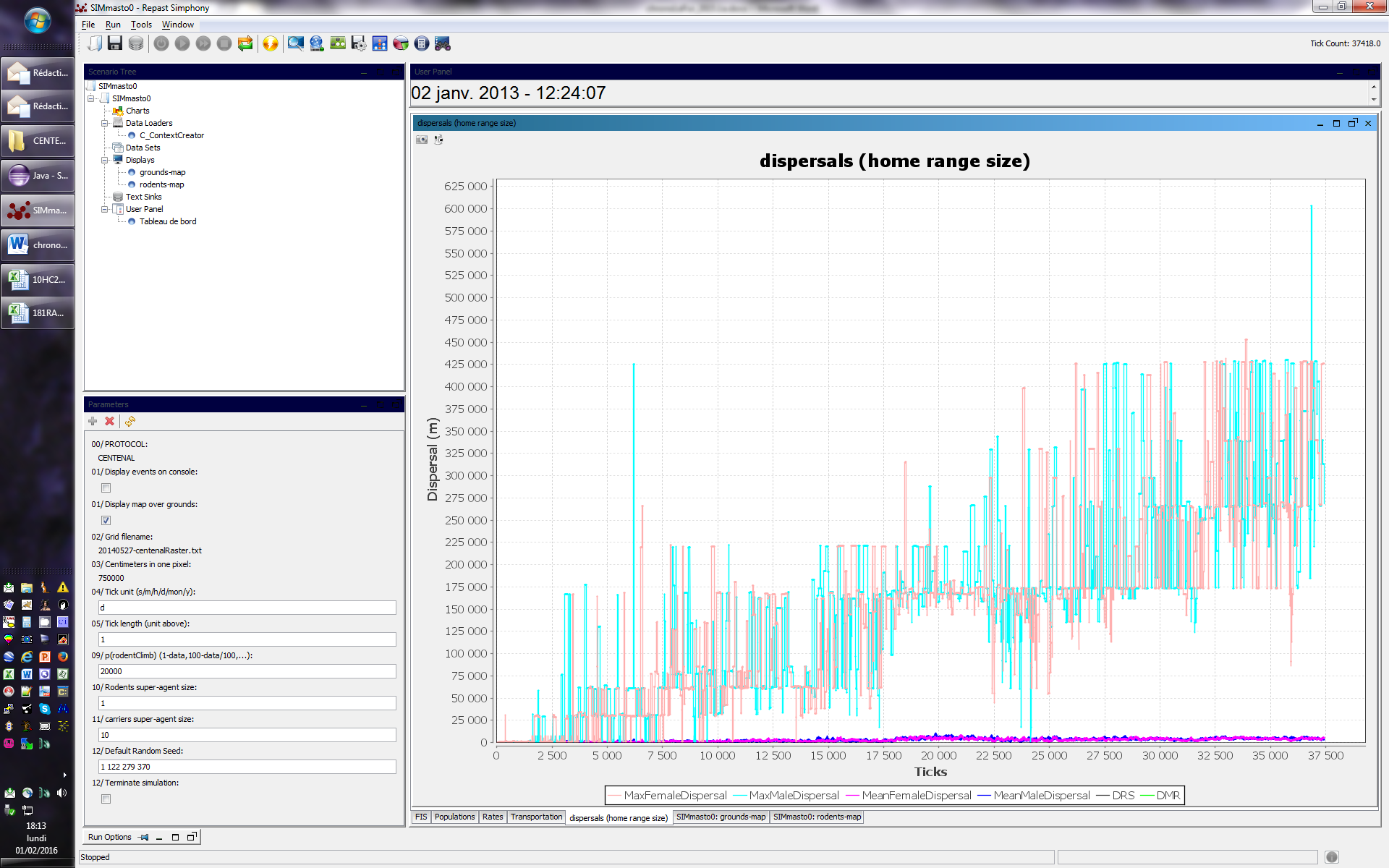


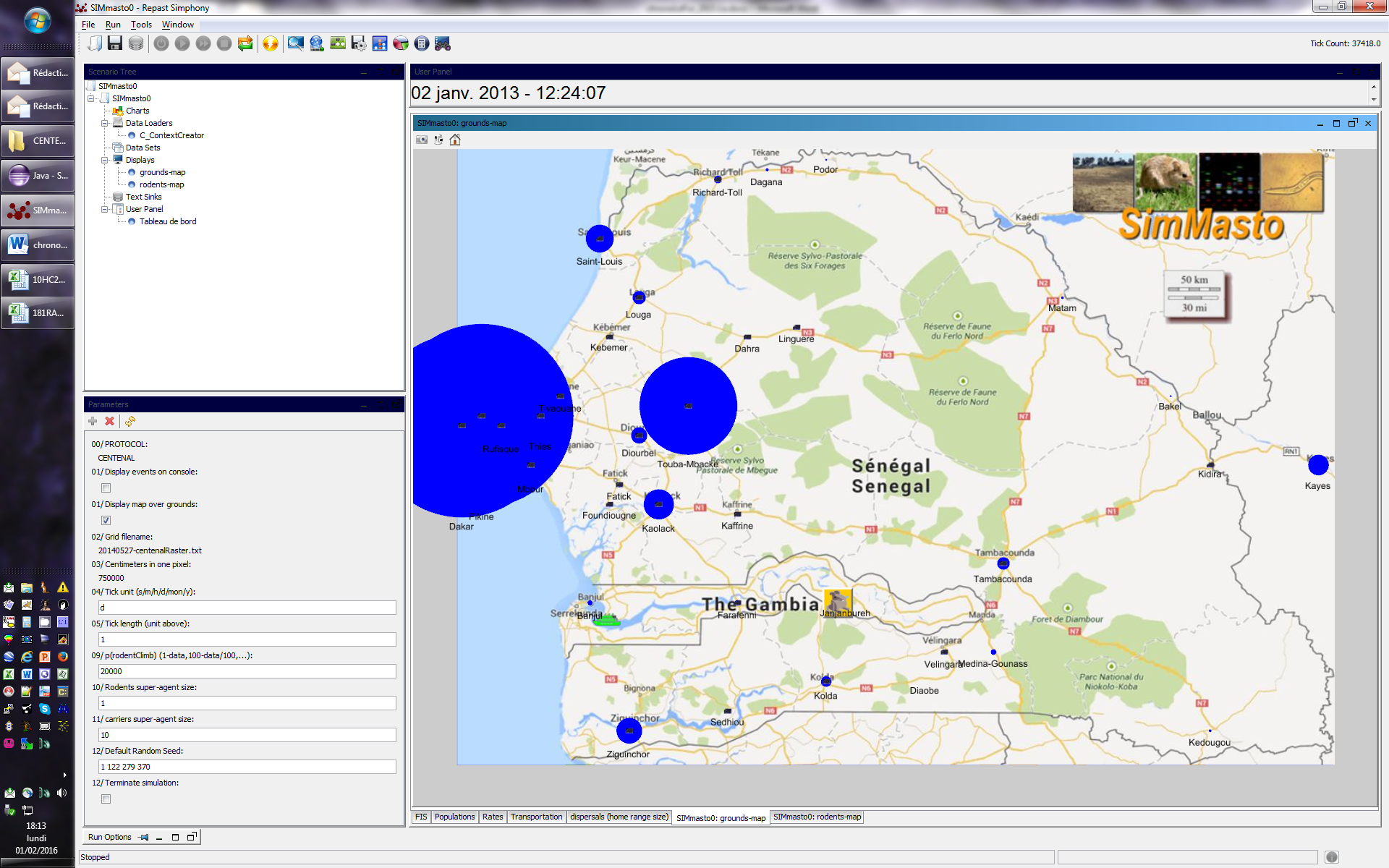


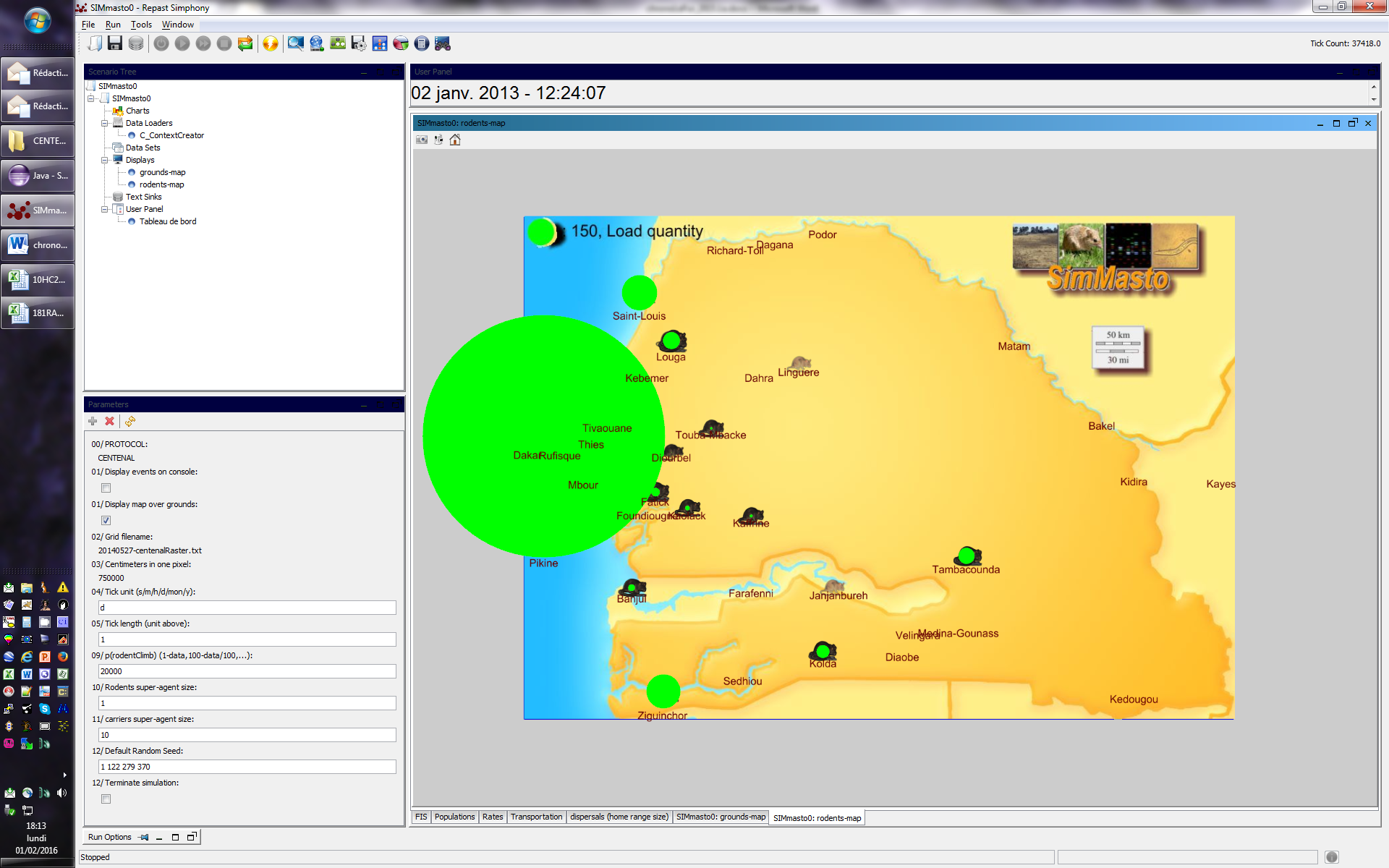














## 16.03.2016 Classification plantes : Différence graminée légumineuses

les Poacées (= Graminées) sont des plantes monocotylédones (un seul cotylédon à la germination), et les Fabacées (anciennement Légumineuses) sont dicotylédones (deux cotylédons à la germination).  
  
Ensuite, les Poacées ont des feuilles généralement étroites, à nervures parallèles. Elles fleurissent sur des épis. Les fleurs sont très réduites, petites et peu visibles, le vent étant le pollinisateur.  
  
Les Fabacées, au contraire, ont des fleurs habituellement colorées, visibles et décoratives, pour être pollinisées par les insectes. Qui plus est, leurs feuilles sont découpées en folioles.

## 29.03.2016 Processus de décision gerbilles

* Si arrivé à destination gerbille : reproduction si arrivé à destination végétation : alimentation
* Perception : vegetation, gerbilles
* Délibération : reproduction nourriture
* Si pas de critères, choisir le plus proche (sauf celui où elle se trouve).

## 02.05.2016 taille de simmasto

Java reverse done

146 classes

571 attributes

314 relations

1071 operations

## 03.06.2016 centenal : vérification de la bonne interaction mâles-femelles

Sur ~1000 cas où il y a seulement deux rongeurs sur un site, moins d’une dizaine où les deux sont compatibles et se reproduisent (pourrait être intéressant de quantifier exactement le taux de réussite

**if** ((**this**.currentSoilCell.getFullLoad\_Urodent() == 2) && (**this**.isFemale() != rodent.isFemale())) System.*err*.println("C\_RodentCommensal.interact() " + **this**.isFemale() + "(" + **this**.getAge\_Uday() + "/" + **this**.readyToMate + "/" + **this**.isSexualMature() + "/" + **this**.isPregnant() + ")" + rodent.isFemale() + "(" + rodent.getAge\_Uday() + "/" + rodent.readyToMate + "/" + rodent.isSexualMature() + "/"+ rodent.isPregnant() + ")");

et :

**if**(**this**.getCurrentSoilCell().getFullLoad\_Urodent()==2)System.*err*.println("A\_Mammal.mateWithMale()"+**this**+" x " +male);

## 29.06.2016 taille de Centre Informations

Java reverse done

24 classes

42 attributes

21 relations

141 operations

## 15.09.2016 Re : Comment augmenter la précision de l’équation linéaire d’une courbe de tendance

Bonjour,  
  
Voilà, après avoir laissé passer cette journée de lendemain de veille… J’ai résolu mon problème.   
  
Il suffit de faire un clique droit sur l’équation dans le graphique (pas sur la courbe) et de sélectionner le format d’étiquette puis en spécifiant le format scientifique on a accès à la précision des décimales.   
  
C’était pourtant si simple… Il y a des journées comme ça!  
  
À la prochaine.

## 03.11.2016 refonte modèle PDE

### A\_Animal Deliberation

/\*\* Second stage of the scheme perception, deliberation action of Ferber 1999 :<br>

\* makes animals systematically interact with others and returns the available soil cells.

\* **@param** perceivedThings TreeSet <I\_situated\_thing> listeVisibleObjects from perception method

\* **@return** candidate targets

\* Version J.E.Longueville & J.Le Fur 2011 / jlefur 03.2012 \*/

**protected** TreeSet<C\_SoilCell> deliberation(TreeSet<I\_SituatedThing> perceivedThings) {

TreeSet<C\_SoilCell> candidateTargets = **new** TreeSet<C\_SoilCell>();

**if** (**this**.hasToDisperse && **this**.lastContainerLeft != **null**) candidateTargets.remove(**this**.lastContainerLeft);

C\_SoilCell currentDest = **null**, testedCell = **null**;

**for** (I\_SituatedThing oneThingPerceived : perceivedThings) {

// if animal encountered, systematically interact - subclasses override interact procedure

**if** (oneThingPerceived **instanceof** A\_Animal && oneThingPerceived != **this**) {

**if** (**this**.interact((A\_Animal) oneThingPerceived)) **break**;

}

// Select soil cells with best affinities

**else** **if** (oneThingPerceived **instanceof** C\_SoilCell) {

testedCell = (C\_SoilCell) oneThingPerceived;

**if** (testedCell.getAffinity() > **this**.getCurrentSoilCell().getAffinity()) {

**if** (currentDest == **null** || testedCell.getAffinity() > currentDest.getAffinity()) {

candidateTargets.clear();

candidateTargets.add(testedCell);

currentDest = testedCell;

}

**else** **if** (testedCell.getAffinity() == currentDest.getAffinity()) candidateTargets.add(testedCell);

}

}

**else** **if** (oneThingPerceived **instanceof** C\_LandPlot) {}// Currently no interaction with landplots

**else** **if** (oneThingPerceived **instanceof** C\_Background) {}// Background has to be managed as an object

**else** **if** (oneThingPerceived **instanceof** C\_Vegetation) {}// Currently no interaction with vegetation

**else** **if** (oneThingPerceived != **this**) A\_Protocol.*event*(

"A\_Animal.deliberation(): neither an Animal/SoilCell/LandPlot/Vegetation/Background"

+ oneThingPerceived.getClass(), *isError*);

}

**return** candidateTargets;

}

### A\_Animal action\_Uday

/\*\* Third stage of the scheme perception, deliberation, decision action in the case of animal - action select and move towards a

\* direction - dedicated to be sophisticated in daughter classes JLF 06.2014 \*/

**public** **void** action\_Uday() {

**if** (!trappedOnBoard) {// if trapped on board, do nothing

// First case: if arrived in wrong place, get back two steps

**if** (**this**.currentSoilCell.getAffinity() <= *DANGEROUS\_AREA\_AFFINITY*) {

**this**.nextMove\_Umeter.x = -2. \* **this**.nextMove\_Umeter.x;// **TODO** JLF 2016.01 number in source get back two steps

**this**.nextMove\_Umeter.y = -2. \* **this**.nextMove\_Umeter.y;// **TODO** JLF 2016.01 number in source get back two steps

**this**.targetPoint\_Umeter = **null**;

}

// Second general case : decision then action (movetoDestination)

**else** selectDestination(deliberation(perception()));

**this**.moveToDestination();

}

}

### A\_Animal step\_Utick

/\*\* Realize action\_Uday, then compute indicator and super

\* **@see** A\_Animal#action\_Uday \*/

@Override

**public** **void** step\_Utick() {

**if** (!**this**.trappedOnBoard) {// if trapped, do nothing - JLF 07.2013

action\_Uday();

computeMaxDispersalDistance\_Umeter();

**super**.step\_Utick();

}

**if** ((**this** **instanceof** C\_Rodent) && !**this**.isDead() && (**this**.currentSoilCell.getAffinity() <= *DANGEROUS\_AREA\_AFFINITY*))

**this**.testDeath(*DANGEROUS\_AREA\_MORTALITY\_RATE*);// **TODO** JLF 2015.04 put in C\_Rodent.step\_Utick

}

### A\_Mammal step\_Utick

/\*\* Mammals' activity within one tick (inherit animal's activity also) - rev JLF 03/07.2014 \*/

/\*\* All methods must be called at each step \*/

@Override

**public** **void** step\_Utick() {

updatePhysiologicStatus();

**super**.step\_Utick();

}

### C\_RodentFossorial action\_Uday

/\*\* Manage activity when rodent is within a burrow system, mature...<br>

\* Adapted from JEL 2010, JLF 03,06.2014 full rev. from svn507, JLF 07.2014 \*/

@Override

**public** **void** action\_Uday() {

**if** (**this**.currentSoilCell.isFull()) **this**.hasToDisperse = **true**;

// within breeding season

**if** (A\_Protocol.*isBreedingSeason*()) {

// pre-mature

**if** (preMature && (isMale() || hasToDisperse)) {

// inside burrow

**if** (**this**.getCurrentSoilCell() **instanceof** C\_BurrowSystem) randomExitOfBurrow();

// outside burrow

**else** **if** (targetedContainer == **null**) **super**.action\_Uday();

**else** moveToDestination();

}

// sexually mature

**else** **if** (isSexualMature() && (isMale() || hasToDisperse)) {

// inside burrow

**if** (**this**.getCurrentSoilCell() **instanceof** C\_BurrowSystem) {

**for** (C\_Rodent accointance : **this**.getCurrentSoilCell().getRodentList())

**this**.interact(accointance);

randomExitOfBurrow();

}

// outside burrow

**else** **if** (targetedContainer == **null**) **super**.action\_Uday();

**else** moveToDestination();

}// else immature hence, do nothing

}

// JLF 10.2016

**if** (((C\_SoilCell) **this**.getCurrentSoilCell()).getAffinityLandPlot() != **null**) {

**if** (preMature && ((C\_SoilCell) **this**.getCurrentSoilCell()).getAffinityLandPlot().isFull()) {

**this**.setHasToDisperse(**true**);

**this**.getNewRandomDisplacement();

**this**.moveToDestination();

}

}

**else** {// outside breeding season

**if** ((**this**.getCurrentSoilCell() **instanceof** C\_BurrowSystem) && **this**.hasToDisperse) randomExitOfBurrow();

**else** **super**.action\_Uday();

}

}

### C\_HumanCarrier step\_Utick

/\*\* Activity within one time step : if a city is reached: select a new city, elaborates its path, unload rodents. If new city not

\* reached : steps towards its current path. If vehicle speed is higher than cell size selectNextNode() is repeated several

\* times / authors P.A.Mboup & J.Le Fur - sept.2012, rev. PAM 03.2014, JLF 10.2015 \*/

@Override

**public** **void** step\_Utick() {

**double** distanceTravelled = 0;

**this**.pathWanderer.intermediateDistanceTravelled = 0;

**while** (**this**.pathWanderer.crossingIntermediateDistance(distanceTravelled, **this**.vehicle.getSpeed\_UmeterByTick())) {

distanceTravelled = 0; // it is because continue of the loop.

// 1) arrived in any path cell towards the targetCell

**if** (**this**.pathWanderer.hasToSetNextNode(**this**.currentSoilCell)) setNextNode();

// 2) reached a destination city (just before stopped)

**else** **if** (**this**.pathWanderer.getPathEnd() && **this**.parked == **false**) {

goParked();

**this**.setDead(parked);

**break**; // pour que parked reste plus de temps true (pendant un step)

}

// 3) just before leaving

**else** **if** (**this**.parked) {

startTraveling();

**continue**;

}

**if** (!**this**.parked) { // safety measure

**double** maxDistanceToMove\_Umeter = (**this**.vehicle.getSpeed\_UmeterByTick() - **this**.pathWanderer.intermediateDistanceTravelled)

/ **this**.pathWanderer.track\_slow\_factor;

**this**.computeNextMoveToTarget(maxDistanceToMove\_Umeter);

distanceTravelled = Math.*sqrt*(**this**.nextMove\_Umeter.x \* **this**.nextMove\_Umeter.x + **this**.nextMove\_Umeter.y

\* **this**.nextMove\_Umeter.y)

\* **this**.pathWanderer.track\_slow\_factor;

**this**.moveToDestination();

}

}

**super**.step\_Utick();

}

## 25.11.2016 Commit 979

refactoring d'à peu près toutes les méthodes d'action en actionXxx

en vue 1) de l'intégration des dépenses énergétiques et 2) de la clarification du modèle PDE (percep-delib-exec)

- création d'un C\_RodentFossorialCommonVole pour chize

Protocoles:

Gerbille: plante car manque les méthodes de fossorial transportées vers CommonVole

Mus Dodel: OK

Mus transport: null pointer dans un graphe de carrier !

Rattus Centenal: démarre en 1940, plante en 1943

Rattus decenal: OK

Bandia: rongeurs ne se piègent plus et bougent mal car manque les méthodes de fossorial transportées vers CommonVole

Cage: OK (enclosure toujours pas, hybrid uniforme: OK)

Chize: OK