**Chrono Gerbil**

**14 09-2015 : début du stage mise en place des outils de travail**

* Installation des logiciels Repast Simphony, Open VPN, SambAgro connector
* Création de :
* l’interface I\_gerbil\_constants
* Ajout du protocole le C\_ContextCreator
* Création de la classe protocole C\_ProtocoleGerbil
* Ajout de Gerbil sur la classe C\_ChooseProtocol

**16-09-2015 : Création d’une fonction qui donne la distance (km) entre deux points géographiques.**

* Utilisation du théorème de Clarke pour transformer les coordonnées géographiques en coordonnées cartésiennes.

**17-09-2015 : Recadrage de la petite emprise**

**18-09-2015 : Création, utilisation d’un premier raster**

* Affichage sur le simulateur
* Utilisation de la procédure colorMapGerbilGrid de la classe C\_Style2dAffinityType

**21-09-2015 : cadrage de la moyenne emprise sur les cartes de pluies**

* Définition du cadre correspondant
* Découpage des cartes de pluies de la moyenne emprise

**22-09-2015 : définition d’une palette de couleur**

* Palette définie pour douze cartes c’est-à-dire pour une période d’une année

**23-09-2015 : Classe C\_ConvertRainFiles qui convertie une carte de pluies en un raster**

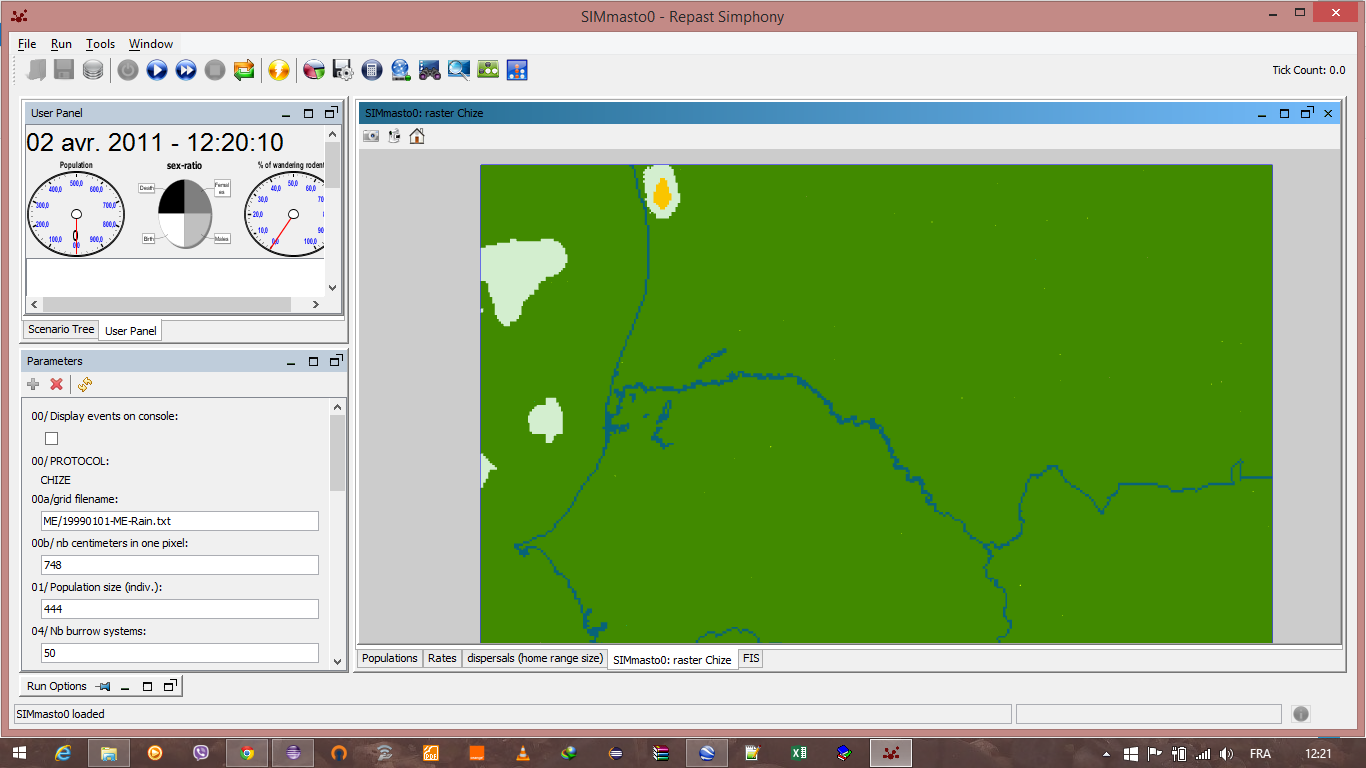
* Redéfinition de la fonction createGround en readBitmapFile
* Sauvegarde de la matrice obtenue sur un fichier de format csv (saveBitmapFile)

**24-09-2015 : Extraction et conversion**

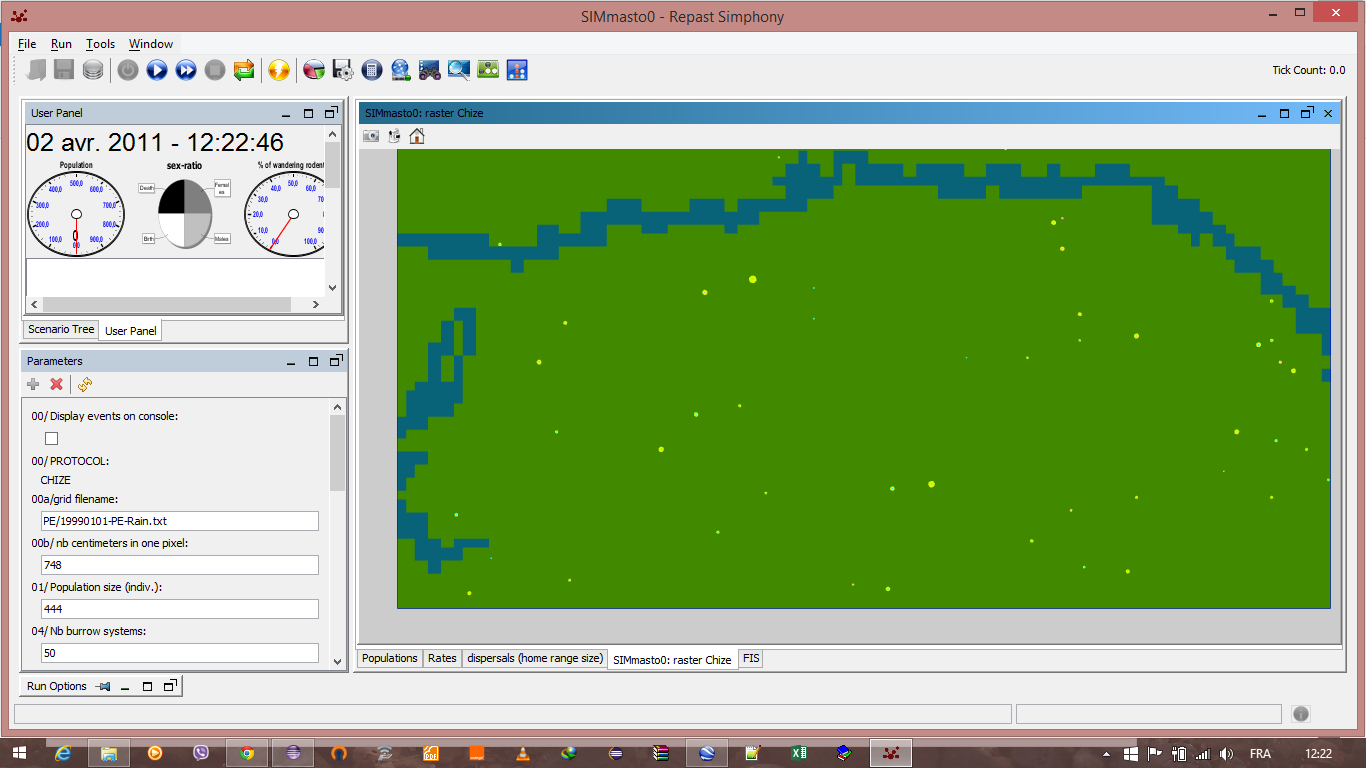
* Création de la liste des noms de fichier contenus dans le dossier Rain (buildFileNameList)
* ME et PE:
  + Lecture de la carte
  + Conversion en ascii
  + Construction du nom de fichier avec la fonction computeFileName
  + Sauvegarde du raster de la moyenne emprise
  + Extraction de la matrice de la petite emprise par extractMatricePE
    - Calcul des coordonnées géographiques de la projection du point Nord-Ouest de la petite emprise sur la hauteur et la largeur de la moyenne emprise
    - Calcul de la distance entre le point Nord-Ouest de la moyenne emprise et les deux projections obtenues dans le point ci-dessus
    - Extraction de la matrice PE
  + Sauvegarde de la matrice PE

**26-09-2015 : Simulation des rasters ME et PE obtenu**

* ME



* PE : extrait à partir de la moyenne emprise



**28-09-2015** : mise au point de la classe C\_GeographyCordinateToCorrespondingCell qui détermine la cellule correspondant aux coordonnées géographiques fournies.

Ainsi, j’utilise la fonction geographyCoordinateToDistance\_Umeter qui calcul la distance entre deux points géographiques en se basant sur la transformation entre systèmes géodésiques de Clarke 1880 IGN, pour déterminer la distance entre les projections du point avec la hauteur et la largeur du raster.

**Procédure de traitement des cartes de pluies de 1990 à 2013**

* Création du masque de la moyenne emprise masqueME.1a.cpt
* Création de la palette de 9 couleurs Contrainte : deux variations de jaune
  + Modèle : TRMM-3B42-19990101-19990131.gif
* Charger la palette de couleur 20150922-palettePluie09Couleurs.xml
* Découper l’image de la moyenne emprise selon le masque ME
* Conserver le nom initial du fichier (date de la carte)
* Ré-échantillonner la carte en pixels (976 x 606 en hauteur).
* Sauver le fichier en bmp

**07-10-2015 : modification de la classe C\_GeographyCordinateToCorrespondingCell**

C\_ GeographyCordinateToCorrespondingCell changée par C\_GeographicCoordinateConverter et optimisation des fonctions :

* distanceToRasterOrigin\_Umeter
* convertCoordinate\_Umeter
* correspondingCell

**07-10-2015 : Constructeur de la classe (C\_Event) avec un paramètre String.**

* Diviser les mots séparés par « ; » puis conversion des différents mots.
* Prise en charge des coordonnées géographiques dans la création d’événement
* Suppression des constructeurs outre que celui décrit ci-dessus
* Ajout de la classe C\_ProtocoleGeographique : qui est l’intermédiaire entre A\_Protocole et les autres classes qui veulent utiliser le convertisseur de coordonnées géographiques (C\_GeographicCoordinateConverter)
* Lecture du chronogramme avec des coordonnées géographiques par SimMasto

**20-10-2015 : Généralisation du convertisseur, mise en place du génome de la gerbille et de la classe végétation.**

**Généralisation du convertisseur :**

* Ajout de la variable geographicCoordinateConverter dans le protocole intermédiaire C\_ProtocoleGeographic :

**public** C\_GeographicCoordinateConverter geographicCoordinateConverter = **null**;

* Initialisation de la variable par tout protocole qui utilise les coordonnées géographiques dans la fonction initProtocole avec les coordonnées géographiques de l’origine.

Exemple : pour la gerbille

**this**.geographicCoordinateConverter=**new**C\_GeographicCoordinateConverter(**new** Coordinate(I\_gerbil\_constants.*gerbilPELongitudeWest\_Udegree*,I\_gerbil\_constants.*gerbilPELatitudeNorth\_Udegree*));

**Mise en place du génome de la gerbille :**

* Création de la classe C\_GenomeGerbillusNigerae qui étend C\_genomeAmiota

**public** C\_GenomeGerbillusNigeriae() { **super**();

alleles.put(*LITTER\_SIZE*, 4.);//source:between 2 and 6 (Sicard and Fuminier, 1996)

alleles.put(*WEANING\_AGE\_Uday*, 26.);//source: Nomao, 2001 (LG comm. Pers.): weaning is three weeks, however young females are not sexually mature between around two months

alleles.put(*MATING\_LATENCY\_Uday*, 39.);//source: **TODO** JLF 10/2015 ask LG

alleles.put(*GESTATION\_LENGTH\_Uday*, 22.);//source: **TODO** JLF 10/2015 ask LG

makeAmniotaBivalent(**this**.alleles);

}

* Création des Gerbilles suivant les données fournies par une évenement: Fonction createRodentWithEvent (C\_Event event)
* IconsSelector :
  + Nom de l’image de Gerbille à afficher IconsSelector : « gnigeriae1 »
  + Renvoi de l’image si le protocole sélectionné est celui de la gerbille

**if**(C\_Parameters.*PROTOCOL*.equals("GERBIL\_PROTOCOL"))

**return** getNameOfImageGerbil(agent);

* + **getColorGerbil : A définir**
* Initialisation du gestionnaire d’image et enregistrement des images qui seront utilisées au cours de la simulation dans le factory :

**public** **void** initGerbil(ShapeFactory2D factory) {

**this**.ELLIPSE\_SCALE = 4.f; IMAGE\_SCALE = .4f;

**this**.IMAGE = **true**; **this**.factory = factory;

selectImg = **new** C\_IconSelector();

**if** (IMAGE) {

factory.registerImage(C\_IconSelector.*GERBIL\_ICON*,selectImg.loadImage(C\_IconSelector.*GERBIL\_ICON*)); }

}

**Classe C\_Vegetation :**

Calcule de la croissance de la végétation par jour avec la formule :

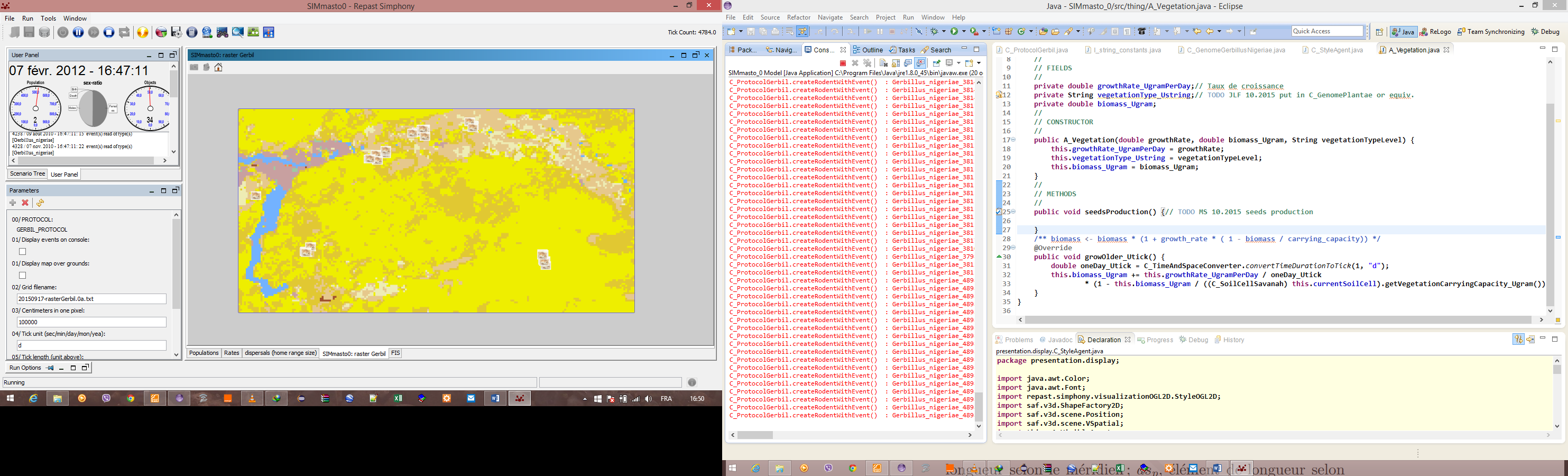
biomass = biomass \* (1 + growth\_rate \* ( 1 - biomass / carrying\_capacity))

Biomasse de plus par jour

pour y arriver, il faut faire le calcul d’un jour par tick(oneDay\_Utick) et divisé la nouvelle biomasse par le oneDay\_Utick :

**double** oneDay\_Utick = C\_TimeAndSpaceConverter.*convertTimeDurationToTick*(1, "d");

**this**.biomass\_Ugram += **this**.growthRate\_UgramPerDay / oneDay\_Utick ;



**24-10-2015 : Conversion de coordonnées continius space en coordonées géographique**

* Utilisation de la règle de trois

public Coordinate convertCell\_Udegree(Coordinate cell, Coordinate width\_HeightOrigin\_Umeter)

**24-10-2015 : Rédaction du mémo pour la conversion des coordonnées géographiques en coordonnées d’espace continue.**

**28-10-2015 : Découpage d’une nouvelle emprise, étude du système gerbille**

* La petite emprise compte 214 cellules en largeur et 110 cellules en hauteur ce qui nous fait un total de 23540 cellules. Ainsi, l’étude de l’interaction de la gerbille avec son environnement tout en conservant les détails (visibilité des objets et des agents) sera mal supporté par le simulateur. En effet, la simulation de la pluviométrie (180 rasters qui équivaut à 15 ans de pluies) et des gerbilles en même temps nous a permis de constater le ralentissement de SimMasto.
* Ce qui était prévu sur la petite emprise c’est de représenter sur la cellule les gerbilles, une chouette, la végétation et un petit cercle qui va représenter la pluviométrie. Ce qui implique qu’il y aura une multiplication du nombre 23540 par le nombre d’objet et d’agent prévu dans une cellule mais aussi une augmentation de la lenteur qui était constaté. C’est ainsi qu’une étude sur une emprise très petite a été envisagée (TPE).
* Extraction d’une emprise 30\*30 qui représente la très petite emprise.
* Etude du système de la gerbille

**09-11-2015 : Extraction de la très petite emprise (buildMatriceTPE)**

* Construction de la liste de nom des fichiers qui sont sur le répertoire data\_raster/rainPE
* Calcul de la position du point d’origine de la très petite emprise par rapport à l’origine de la petite emprise
* Construction de la matrice du raster de la petite emprise
* Extraction de la matrice de la très petite emprise
* Construction et sauvegarde du raster de la très petite emprise en fichier .txt sur le répertoire data\_raster/rainTPE

Code de la fonction **buildMatriceTPE**

/\*\*Extract the very little grip matrix in the average grip little matrix \*/

**public** **void** buildMatriceTPE(Coordinate newOrigineCoord\_Udegree,Point dimensionRaster) {

**int**[][] matrixTPE = **new** **int**[ dimensionRaster.x][dimensionRaster.y];

ArrayList<String> fileNamePElist = buildFileNameList(*RASTER\_PATH* + "/rainPE");

**int** computerFile = 0;

**int**[][] matrixPE;

C\_ConvertGeographicCoordinates converter = **new** C\_ConvertGeographicCoordinates(**new**Coordinate(I\_gerbil\_constants.*gerbilPELongitudeWest\_Udegree*, I\_gerbil\_constants.*gerbilPELatitudeNorth\_Udegree*));

Coordinate correspondingCell\_Ucs = converter.convertCoordinate\_Ucs(newOrigineCoord\_Udegree.x, newOrigineCoord\_Udegree.y);

**int** positionWidth = (**int**) correspondingCell\_Ucs.x;

**int** positionHeight = (**int**) correspondingCell\_Ucs.y;

System.*err*.println("C\_ConvertRainFiles.buildMatriceTPE() : position largeur " + positionWidth + " position hauteur " + positionHeight);

**for** (String fileName : fileNamePElist) {

fileName = fileNamePElist.get(computerFile);

matrixPE = extractMatricePEInFileTxt(*RASTER\_PATH* + "/rainPE/" + fileName);

**if** ((positionHeight + dimensionRaster.x > matrixPE[0].length) || positionWidth + dimensionRaster.x > matrixPE.length) {

System.*out*.println("C\_ConvertRainFiles.buildMatriceTPE() : Extraction Impossible !!!");

}

**else** {

**for** (**int** i = positionHeight+dimensionRaster.x ; i > positionHeight; i--)

**for** (**int** j = positionWidth; j < positionWidth + dimensionRaster.y; j++)

matrixTPE[j - matrixPE[j][matrixPE[0].length - (i-1)];

**this**.saveBitmapFile(matrixTPE, **this**.computeFileName(fileName, "TPE"), "TPE");

computerFile ++;

}

}

}

**10-11-2015 jusqu’au 06-12-2015: Interruption à cause de la reprise des cours du master.**

**07-12-2015 : Discussion sur Skype sur la fonction d’extraction de la TPE avec J. LeFur**

**09-12-2015 : C\_ZoomExtractor :**

* Extraction de la très petite emprise sur les fichiers de la petite emprise en format txt.
  + Calcul de la distance du point d’origine de la zone à extraire
  + Extraction du raster sur les fichiers txt
  + Enfin, extraction de la très petite emprise

Code correspondant :

/\*\* Extract the very small grip matrix in the grip little matrix \*/

ArrayList<String> fileNamePElist = convertFile.buildFileNameList(*RASTER\_PATH* + "/rainPE");

**int** Computer = 0;

System.*err*.println("C\_ConvertRainFiles.buildMatriceTPE() : position largeur " + positionOriginTPE.x + " position hauteur "

+ positionOriginTPE.y);

**for** (String fileName : fileNamePElist) {

fileName = fileNamePElist.get(Computer);

matrixPE = extractMatricePEInFileTxt(*RASTER\_PATH* + "/rainPE/" + fileName);

**if** ((positionOriginTPE.y + zoomHeight > matrixPE[0].length) || positionOriginTPE.x + zoomLength > matrixPE.length) {

System.*out*.println("C\_ConvertRainFiles.buildMatriceTPE() : Extraction Impossible !!!");

}

**else** {

**for** (i = positionOriginTPE.y + zoomHeight; i > positionOriginTPE.y; i--)

{

**for** (j = positionOriginTPE.x; j < positionOriginTPE.x + zoomLength; j++) {

matrixTPE[j - positionOriginTPE.x][(positionOriginTPE.y + zoomHeight) - i] = matrixPE[j][matrixPE[0].length - (i - 1)];

}

}

fileNameTPE = fileName.split("-")[0] + "-TPE-Rain";

convertFile.saveBitmapFile(matrixTPE, fileNameTPE, "TPE", repertory);

Computer++;

}

}

* Extraction de la très petite emprise sur les fichiers de moyenne emprise en format bmp.
  + Conversion de la bmp en raster
  + Calcul de la position de l’origine de la zone à extraire par rapport à l’origine de la moyenne emprise.
  + Composition du nom de fichier à enregistrer
  + Sauvegarde de la zone extraite

Code correspondant

/\*\* Extract the very small grip matrix in the average grip matrix \*/

ArrayList<String> fileNameMEList = convertFile.buildFileNameList(*RASTER\_PATH* + "rainFiles");

**for** (String fileNameInRain : fileNameMEList) {

fileNameInRain = fileNameMEList.get(computer);

matrixME = convertFile.readBitmapFile("rainFiles/" + fileNameInRain);

**if** ((positionOriginTPE.y + zoomHeight > convertFile.dim\_Ucell.height)

|| positionOriginTPE.x + zoomLength > convertFile.dim\_Ucell.width) {

System.*out*.println("C\_ConvertRainFiles.buildMatriceTPE() : Extraction Impossible !!!");

}

**else** {

**for** (i = positionOriginTPE.y + zoomHeight; i > positionOriginTPE.y; i--) {

**for** (j = positionOriginTPE.x; j < positionOriginTPE.x + zoomLength; j++) {

matrixTPE[j - positionOriginTPE.x][i - (positionOriginTPE.y + 1)] = matrixME[j][i];

}

}

fileNameTPE = (fileNameInRain.split("-")[2]).substring(0, 6) + "-TPE-Rain";

convertFile.saveBitmapFile(matrixTPE, fileNameTPE, "TPE", repertory);

omputer++;

}

}

**10 au 23-12-2015 : C\_ZoomExtractor**

* Finalisation de la fonction C-Zoomextractor (extraction sur les fichiers txt et bmp)

**private** **void** extractZoom(String reference\_Ustring, Coordinate zoomOrigin\_Udegree, **int** zoomLength\_Ucell, **int** zoomHeight\_Ucell,

String pathRecording, String pathFolder) {

C\_ConvertRainFiles convertFile = **new** C\_ConvertRainFiles();

**int**[][] matrixTPE = **new** **int**[zoomLength\_Ucell][zoomHeight\_Ucell];

**int**[][] matrixOrigine = **null**;

String fileNameTPE = "";

**int** computer = 0;

Point positionOriginTPE\_Ucell = positionOriginPoint(zoomOrigin\_Udegree, reference\_Ustring);

System.*err*.println("C\_ZoomExtractor.extractZoom() : position width " + positionOriginTPE\_Ucell.x + " position heigth " + positionOriginTPE\_Ucell.y);

ArrayList<String> fileNameList = convertFile.buildFileNameList(*RASTER\_PATH* + pathFolder);

**if** (fileNameList.isEmpty()) {

System.*out*.println("C\_ZoomExtractor.extractZoom(): Répertoire " + pathFolder + " est vide!!! ");

**return**;

}

**int** Computer = 0;

String format;

**switch** (reference\_Ustring) {

**case** "PE" : {

// Extract the very small grip matrix in the grip little matrix

**for** (String fileName : fileNameList) {

fileName = fileNameList.get(Computer);

format = getExtension(fileName);

**switch** (format) {

**case** "txt" : {

matrixOrigine = extractMatrixPEInFileTxt(*RASTER\_PATH* + pathFolder + "/" + fileName);

fileNameTPE = fileName.split("-")[0] + "-TPE-Rain";

}

**break**;

**case** "bmp" : {

matrixOrigine = convertFile.readBitmapFile(pathFolder + "/" + fileName);

fileNameTPE = "TPE-landcoverGerbil";

}

**break**;

}

matrixTPE = extractZoomFromGrid(matrixOrigine, positionOriginTPE\_Ucell, zoomHeight\_Ucell, zoomLength\_Ucell, format);

convertFile.saveBitmapFile(matrixTPE, fileNameTPE, "TPE", pathRecording);

Computer++;

}

}

**break**;

**case** "ME" : {

// Extract the very small grip matrix in the average grip matrix

**for** (String fileName : fileNameList) {

fileName = fileNameList.get(computer);

format = getExtension(fileName);

**switch** (format) {

**case** "bmp" : {

matrixOrigine = convertFile.readBitmapFile(pathFolder + "/" + fileName);

matrixTPE = extractZoomFromGrid(matrixOrigine, positionOriginTPE\_Ucell, zoomHeight\_Ucell, zoomLength\_Ucell, "bmp");

fileNameTPE = (fileName.split("-")[2]).substring(0, 6) + "-TPE-Rain";

convertFile.saveBitmapFile(matrixTPE, fileNameTPE, "TPE", pathRecording);

}

**break**;

}

computer++;

}

**break**;

}

}

}

/\*\* Referral extension of the input file \*/

**public** String getExtension(String fileName) {

**return** fileName.substring(fileName.lastIndexOf(".") + 1);

}

/\*\* Extract TPE matrix in matrix Origin \*/

**public** **int**[][] extractZoomFromGrid(**int**[][] matrixOrigin, Point positionOriginTPE\_Ucell, **int** zoomHeight\_Ucell, **int** zoomLength\_Ucell, String format) {

**int**[][] matrixTPE = **new** **int**[zoomHeight\_Ucell][zoomLength\_Ucell];

**try** {

**for** (**int** i = positionOriginTPE\_Ucell.x; i < positionOriginTPE\_Ucell.x + zoomHeight\_Ucell; i++) {

**for** (**int** j = positionOriginTPE\_Ucell.y; j < positionOriginTPE\_Ucell.y + zoomLength\_Ucell; j++) {

**switch** (format) {

**case** "bmp" :{

matrixTPE[i - positionOriginTPE\_Ucell.x][j - positionOriginTPE\_Ucell.y] = matrixOrigin[i][j];

System.*out*.print(matrixOrigin[i][j] + " ");

}

**break**;

**case** "txt" : {

matrixTPE[i - positionOriginTPE\_Ucell.x][j - positionOriginTPE\_Ucell.y] = matrixOrigin[i][j];

System.*out*.print(matrixOrigin[i][j] + " ");

}

}

}

}

} **catch** (Exception e) {

System.*err*.println("C\_ZoomExtractor.extractByEmprise() : Impossible Extraction!!!");

}

**return** matrixTPE;

}

/\*\* Compute position of TPE origin point \*/

**public** Point positionOriginPoint(Coordinate zoomOrigin\_Udegree, String emprise) {

Point positionPoint;

C\_ConvertGeographicCoordinates converter;

**if** (emprise.equals("ME")) {

converter = **new** C\_ConvertGeographicCoordinates(**new** Coordinate(I\_ConstantGerbil.*gerbilMELongitudeWest\_Udegree*,

I\_ConstantGerbil.*gerbilMELatitudeSouth\_Udegree*));

}

**else** {

converter = **new** C\_ConvertGeographicCoordinates(**new** Coordinate(I\_ConstantGerbil.*gerbilPELongitudeWest\_Udegree*,

I\_ConstantGerbil.*gerbilPELatitudeSouth\_Udegree*));

}

Coordinate correspondingCell\_Ucs = converter.convertCoordinate\_Ucs(zoomOrigin\_Udegree.x, zoomOrigin\_Udegree.y);

positionPoint = **new** Point((**int**) correspondingCell\_Ucs.x, (**int**) correspondingCell\_Ucs.y);

**return** positionPoint;

}

* . Rédaction du mémo sur l’extraction du zoom TPE

**04-01-2016 : Simulation de la très petite emprise MAJ**

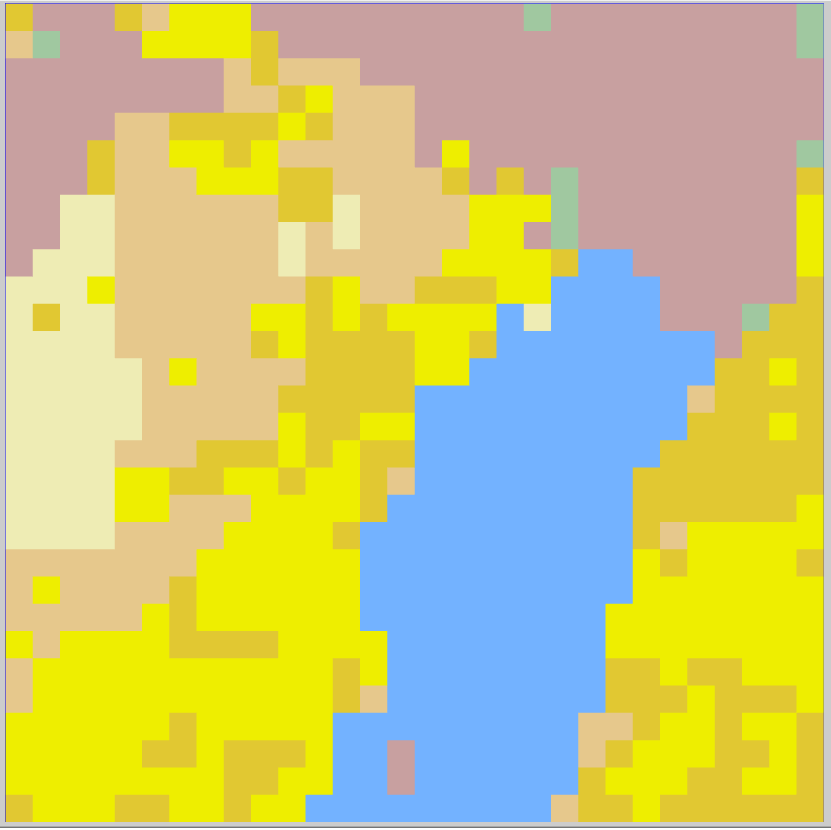
****

Figure 4: Landcover de la TPE simulée

**05-01-2016 : Ajout de repast.simphony.action.display\_RainInSoilCellSavanna et gestion du chrono pour la TPE**

* Affichage de l’évolution de la pluviométrie durant la simulation
* Gestion des événements qui s’opèrent en dehors de la TPE(les ignorés)
* Mise à jour de classe d’extraction et de conversion (C\_ZoomExtractor, C\_ConvertrainFile, C\_ConvertgeographicCoordinate)

**06-01-2016 : Mortalité très rapide des gerbilles**

* Redéfinition de la fonction initFixedParameters et affectation d’une valeur maximale à l’âge d’une gerbille

**public** **void** initFixedParameters() {

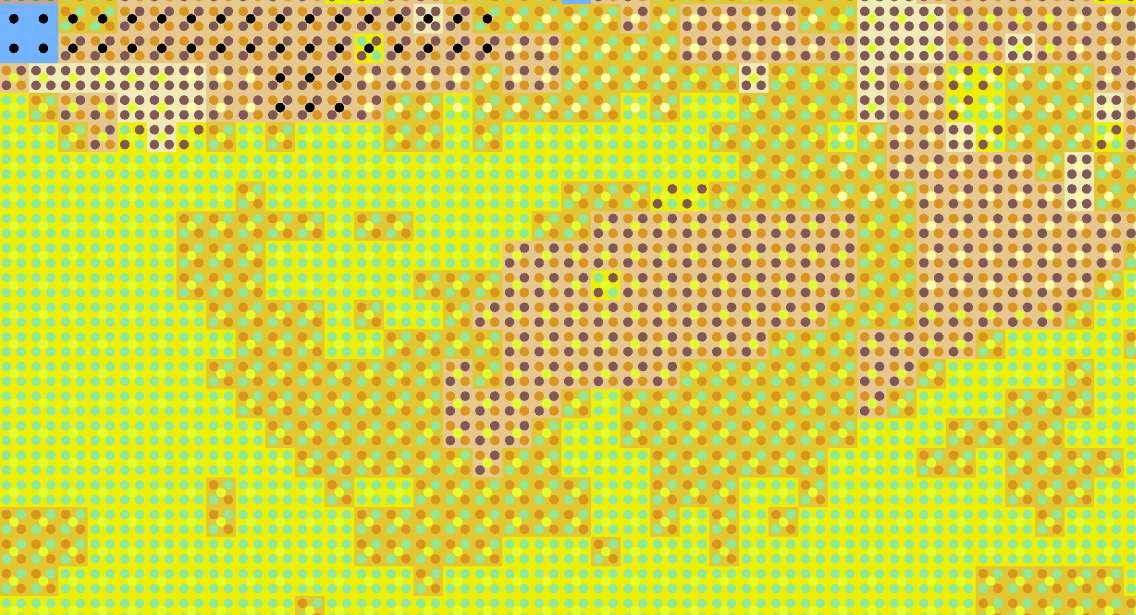
**super**.initFixedParameters();

C\_Parameters.*MAX\_AGE\_Uday* = *gerbilMaxAge*;

}

**07 au 12-01-2016 : Mise en place des objets de la végétation sur les SoilCellSavanna :**

* affichage de quatre objets de végétation avec des couleurs différentes mais également avec des images correspondant au type de végétation

****

**Pour simuler la PE : remplacer,**

* rainUrl = "Zoom\_001\_08.12.2015/" **par** rainUrl = "rainPE/" dans l’interface I\_ConstantGerbille
* Et dans C\_ProtocolGerbil :
* **case** *GERBIL\_EVENT* : {

**if** ((coordinate\_Ucs.x < *widthTPE*) && (coordinate\_Ucs.y < *heightTPE*)

**par**

**if** ((coordinate\_Ucs.x < *widthPE*) && (coordinate\_Ucs.y < *heightPE*)

* **case** *Rain : ”* *-TPE-Rain.txt”* **par** *”* *-PE-Rain.txt”*
* **this**.geographicCoordinateConverter = **new** C\_ConvertGeographicCoordinates(**new** Coordinate(I\_ConstantGerbil.*gerbilTPELongitudeWest\_Udegree*,

I\_ConstantGerbil.*gerbilTPELatitudeSouth\_Udegree*));

**par**

* **this**.geographicCoordinateConverter = **new** C\_ConvertGeographicCoordinates(**new** Coordinate(I\_ConstantGerbil.*gerbilPELongitudeWest\_Udegree*,

I\_ConstantGerbil.*gerbilPELatitudeSouth\_Udegree*));

Et finalement, dans parameters\_scenario\_GERBIL\_PROTOCOL.txt changer la ligne 22 :

type="java.lang.String" defaultValue="20151223-rasterGerbil.0b.txt"

**par**

type="java.lang.String" defaultValue="20150917-rasterGerbil.0a.txt"

**13-01 au 02-02-2016 : Suppression de initFixedParameters ; Croissance de la végétation ; mise en place de la classe C\_RodentGerbil, interaction, déplacement et fabrication de terrier pour chaque ajout de gerbille**

**Suppression de initFixedParameters :**

* Suppression de initFixedParameters sur le protocole gerbille et initialisation des paramètres sur le fichier ParametersScenarion.xml
* Ajout de la variable vegetationList pour stocker tous les objets de végétation contenus dans le raster

**Croissance de la végétation :**

* Mise en place provisoirement des taux de croissance initiale pour chaque type de végétation, du taux de croissance apporté par la quantité de plus déversée sur la cellule et du rayon de perception de la gerbille pour un objet de végétation (voir I\_ConstantGerbil
* Calcul du taux de croissance par jour suivant la quantité de pluies présent dans la cellule
* Mise en place de fonction de vérification du contenue de la cellule en biomasse :

**public** **boolean** isFullBiomass()

* Calcul de la croissance de la biomasse pour chaque pas de temps :
  + Comparaison de la quantité de pluies présent dans la cellule, si une différence est constatée, alors calcul de la nouvelle biomasse par rapport à la nouvelle valeur de la pluviométrie.
* Visualisation de la croissance de la végétation dans le display de la pluviométrie :
  + Ajout du fichier xml thing.ground.C\_Vegetation.style\_gerbil.xml sur le répertoire
  + Prise en compte du fichier de style sur le display Rain-map

<entry>

<string>thing.C\_Vegetation</string>

<string>thing.ground.C\_Vegetation.style\_Gerbil.xml</string>

</entry>

**Classes  C\_RodentGerbil, interaction, déplacement et fabrication de terrier pour chaque ajout de gerbille :**

* Héritage sur la classe C\_RodentFossorial
* Déclaration de la variable vegetationPerceivedList qui reçoit la liste des végétations perçues par la gerbille
* Redéfinition de la fonction de perception : Ajout de la perception des objets de végétation pour un rayon de perception de 280 m
  + Les 280 m représentent : un rayon de couverture de 250m de végétation et le rayon de perception de 30m de la gerbille
* Mise en place de la fonction de recherche de terrier dans la cellule sur un rayon de 30 m autour de la gerbille :

**public** TreeSet<C\_BurrowSystem> getBurrowInCurrentSoilCell()

* Mise en place de la fonction de création de terrier dans un endroit dont les coordonnées sont données en paramètre :

**public** **void** digBurrowSystemAroundRodent(**double** [] newLocation)

**22-02 au 25-02-2016 : Interaction de la gerbille avec la végétation, modification**

* Déprédation de la végétation (CROP, GRASS) avec la diminution de la biomasse
* Interdiction de créer un terrier sur une zone d’eau
* Prise en compte des saisons de reproduction qui dépassent le mois de décembre
* Positionnement des gerbilles retrouvées dans les pelotes de chouette

**29-02-2016 : Changement des couleurs d’un objet qui croit dans le display**

* Définition des fonctions getRedColor, getGreenColor, getBlueColor dans la classe C\_Vegetation
* Appel de ces fonctions respectivement dans les balises redMethodName, greenMethodName, blueMethodName du fichier xml thing.ground.C\_Vegetation.style\_Gerbil pour changer les couleurs selon le type de végétation

**01-03-2016 : Affichage du display rain-map avec un valueLayer 2**

* Déclaration d’une nouvelle variable rainGridValueLayer dans la classe C\_LandscapeSolCellSavanna pour le background de rain-map
* Initialisation de la variable createGround :

**this**.rainGridValueLayer = **new** GridValueLayer(*proj\_gridvalue2*, **true**, **new** repast.simphony.space.grid.WrapAroundBorders(),**this**.dimension\_Ucell.width, **this**.dimension\_Ucell.height);

**for** (**int** i = 0; i < **this**.dimension\_Ucell.width; i++) {

**for** (**int** j = 0; j < **this**.dimension\_Ucell.height; j++) {

rainGridValueLayer.set(matriceLue[i][j], i, j);

}

}

* Changement des valeurs du rainGridValue dans la fonction manageOneEvent de la classe C\_ProtocolGerbil :

**for** (**int** i = 0; i < imax; i++) {

**for** (**int** j = 0; j < jmax; j++) ((C\_LandscapeSahelianSavanna) **this**.landscape).setRainGridValueLayer(matriceLue[i][j], i,j);

}

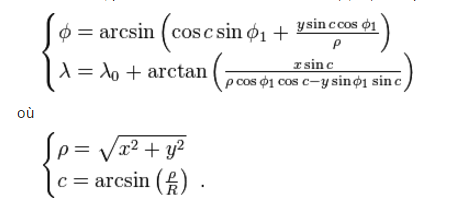
}

* Appel de rainGridValueLayer dans la balise valueLayers du fichier repast.simphony.action.display\_RainSoilCellSavanna.xml

<valueLayers> string>valuegrid2</string> </valueLayers>

## 08/04/2016 : Conversion de coordonnées cartésiennes (espace continu du simulateur) en coordonnées géographique

Utilisation de la formule de projection orthographique pour convertir les coordonnées de l’espace continu en coordonnées géographiques.



* Rayon de la terre suivant l’équateur : a = 6 378 249,2 m
* Rayon de la terre suivant les pôles : b = 6 356 515,0 m

L’excentricité : e =

Avec Φ1 et λ0 les coordonnées géographiques du point pris comme origine du raster, x et y les coordonnées cartésiennes du point à convertir et R le rayon de la terre calculé suivant la latitude Φ1.

Traduction de la formule en code sur C\_ConvertgeographicCoordinates avec la fonction convertCell\_Udegree.

# C:\Users\Administrateur\Desktop\barnowl_tcm9-18232.jpgMaster Recherche

## 20.10.2016 : Gerbillus simulation

Mise en place d’un raster de 9 cellules

Gestion de délibération de la gerbille

Changement de la variable Cell\_Size de I\_ConstantNumeric en ArrayList

Interaction de la gerbille avec les chouettes :

Si une chouette est aperçue par la gerbille alors elle va chercher à se cacher sous un terrier déjà creusé.

Si elle n’en trouve pas, elle cherche un arbuste, se déplacer jusqu’à cet arbuste, crée un terrier avant de se terrer.

Sinon elle se disperse dans la nature.

Ajout du génome de la maturité sexuelle dans la classe C\_genomeEucaryote :

**protected** **static** **final** ArrayList<Double> *SEXUAL\_MATURITY\_MAPLOCS* = **new** ArrayList<Double>(Arrays.*asList*(17.));

**protected** **static** **final** **int** *SEXUAL\_MATURITY\_LOCUS* = 0;

Initialisation de l’allèle sur le constructeur du Génome

**this**.alleles.put(*SEXUAL\_MATURITY\_Uday*, 1.);

Utilisation sur la classe C\_GenomeAmniota :

**this**.xsomePairAmniota.setGenePairAtLocus(*SEXUAL\_MATURITY\_LOCUS*, **new** C\_Gene(allelesMap.get(*SEXUAL\_MATURITY\_Uday*) + getTaxonSignature(), *SEXUAL\_MATURITY\_MAPLOCS*.get(0), mutator, **this**.xsomePairAmniota.xsomeStrands[0].myId),

**new** C\_Gene(allelesMap.get(*SEXUAL\_MATURITY\_Uday*) + getTaxonSignature(), *SEXUAL\_MATURITY\_MAPLOCS*.get(0), mutator,

**this**.xsomePairAmniota.xsomeStrands[1].myId));

Création de la classe C\_BarnOwl :

Initialisation de sa vitesse de déplacement par jour à 2500 m

## 04.01-2017 : Délibération et action de la chouette et de la gerbille

* Deliberation et action de la chouette :

La chouette perçoit les rongeurs non terrées avec un rayon de perception de 104,167m par pas de temps (1heure)

Ainsi, pendant les heures de chasse comprises entre 19h et 6h du matin, suivant sa quantité d’énergie restant et l’abondance des rongeurs, elle se nourrit et retourne sur son nid dans les heures restantes.

* Délibération et action de la gerbille

La gerbille perçoit les agents et objets sur un rayon de 30m.

Ainsi, par ordre de priorité elle effectue sa délibération.

D’abord, si une chouette est aperçue dans les environs et qu’elle ne soit pas dans un terrier alors elle tente une action de fuite en cherchant un terrier dans les environs. Si un terrier n’est pas aperçu alors, elle se rabatte sur les buissons constitués par des arbustes et si aucun des deux n’est perçu alors elle tente un dernier manœuvre en allant tout droit sans s’arrêter.

Ensuite si aucune chouette n’est perçue alors si le besoin de manger se fait sentir et si la nourriture est disponible alors elle se déplace sur la couverture végétale et se nourrit.

Noter que chaque action menée par les agents engendre une dépense ou un gain d’énergie.

Correction : vérifier si le rongeur n’est pas dans un terrier avant de le déplacer.

La gestion de l’activité des agents suivant le créneau horaire.

## 08032017 : pour changer le background il faut changer la valeur du valueLayers du fichier xml

## 23-03-2017 : SVN 1000

Ajout de six vegetations sur une cellule, override de la fonction getCarringCapacity sur C\_SoilCellSavanna, correction de la perception des chouettes, gestion des heures d'activités de la chouette (à revoir), modification de classe rodent gerbille( deliberation et action)

## 17/05/2017 : SVN 1005

C\_RodentGerbil : vérification de currentSoilCell avant la recherche de terrier, Suppression de l'activitation du step des inspecteurs sur les protocoles.

A\_Protocol : bouclage sur la liste des inspecteurs.

C\_InspectorVegetation : récupération de la matrice de cellules dans la classe C\_InspectorVegetation et création de son constructeur avec comme argument, la matrice citée ci-dessus.

Calcul de la moyenne de la pluviométrie par pas de temps.

Utilisation de la fonction indicateursStoreValues pour sauvegarder les résultats de biomasse totale et la moyenne de la pluviométrie

Entretien J.Marc et Laurent sur la simulation de la souris sur l’ensemble du Sénégal

## 18/05/2017 : anciennes version des parameters.xml

|  |  |  |  |
| --- | --- | --- | --- |
| **Protocole** | **Speed** | **sensing** | **Espèce** |
| Gerbil | 30 | 30 | Gerbil |
| Cages | 5 | 1 | Masto |
| Bandia | 1 | 15 | Masto |
| Enclosure | 2 | 150 | Masto |
| HybridUniform | 3 | 5 | Masto |
| Chize | 2 | 30 | Microtus |
| Chize2 | 2 | 30 | Microtus |
| Dodel | 1 | 5 | Souris |

Suppression du champ parameters pour le speed\_UmeterByDay sur les fichiers xml des prototcoles.

<parameter name="AGENT\_SPEED\_UmeterByDay" displayName="08/ Agent speed (m/tick)"

type="java.lang.Integer" defaultValue="1" isReadOnly="false"

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

Suppression du champ de parameters Agent\_Perception\_radius\_UmeterByday sur les fichiers xml des protocole

<parameter name="AGENT\_PERCEPTION\_RADIUS\_UmeterByDay" displayName="07/ Agent sensing (meters)"

type="java.lang.Integer" defaultValue="15" isReadOnly="false" converter="repast.simphony.parameter.StringConverterFactory$IntConverter"/>

Initialisation du speed\_UmeterByDay dans la classe C\_GenomeAnimalia

Généralisation du gène de la maturité sexuelle

Conversion automatique des speeds et du sensing au passage d’une unité de pas de temps à un autre.

Création du setteur setTrappedOnBoard

Généraliser le readUserParameters de A\_Protocol

Ajout du paramètre Female\_Burrow\_CarringCapacity

Correction des bugs dus aux paramètres sur le protocole fossorial

## Récapitulation des figures à présenter à pdb

# Questions et Opinions

Est-ce que le besoin de s’accoupler entraine une recherche permanente de partenaire ? oui

Est-ce que le manque d’énergie est prioritaire à l’instinct de survie (si un prédateur est perçu) ? non

Si le seuil d’énergie minimum est atteint, faire mourir le rongeur. non

## A faire :

Homogénéiser tous les génomes avec sensing et speed fait

Faire la conversion d’échelle temporelle généralisée fait pas testé

Vérifier gène maturity fait pas testé

Vérifier les protocoles

Bilan des valeurs des traits de vie à partir tableau excel JLF-LG

Mettre max age dans gènes fait

## COMMIT SVN 1011

Modifications en cours

Généraliser le readUserParameters de A\_Protocol + simplification

Suppression du champ parameters pour le speed\_UmeterByDay sur les fichiers xml des protocoles.

Suppression du champ de parameters Agent\_Perception\_radius\_UmeterByday sur les fichiers xml des protocoles

Initialisation de speed\_UmeterByDay dans la classe C\_GenomeAnimalia

Généralisation du gène de la maturité sexuelle

Conversion automatique des speeds et du sensing au passage d’une unité de pas de temps à un autre.

Création du setteur setTrappedOnBoard

Ajout du paramètre Female\_Burrow\_CarringCapacity

Chize cohérent, centenal plante

## 24/05/2017 A FAIRE

* Nom des gerbilles si sexe imposé fait

String rodentName = one\_rodent.getThisName();

one\_rodent.setThisName(rodentName.substring(0,1)+sex+rodentName.substring(2));

* Fusionner les event addThingWithEvent
* Récapituler les pdf cités dans les sources dans doc/biblio
* Vérifier energy\_Ukcal dans chaque procédure actionXXX
* Si gerbille sexuellement active priorité = interact(gerbille) même si energie négative  
  priorité 1 : fuir / 2 : se reproduire

### Déclenchement de la saison de reproduction

* Si végétation en croissance -> 6-methoxybenzaolinone +
* Si rongeur mange végétation avec 6-methoxybenzaolinone -> en rut ok
* Si rongeur mange végétation sans 6-methoxybenzaolinone -> pas en rut

### Gestion de l’énergie : une action =1kcal

## Code gardé :

calcul de coordonnées aléatoire sur une distance d’un kilomètre pour l’ajout des gerbilles trouvées sur un lot de pelote

// Randomly add Gerbillus nigeriae found in owl pellets over a distance of between 0 and 2.5 km

**for** (**int** i = 0; i < Integer.*parseInt*(event.value2); i++) {

coordinateCell.x = event.whereX\_Udouble + ((-1 + (2 \* C\_ContextCreator.*randomGeneratorForInitialisation*.nextDouble()))

\* ***averageDistanceHuntingOWL\_Umeter***) / C\_Parameters.*UCS\_WIDTH\_Umeter*;

coordinateCell.y = event.whereY\_Udouble + ((-1 + (2 \* C\_ContextCreator.*randomGeneratorForInitialisation*.nextDouble()))

\* ***averageDistanceHuntingOWL\_Umeter***) / C\_Parameters.*UCS\_WIDTH\_Umeter*;

**if** (coordinateCell.x < 0) coordinateCell.x = 0.0;

**if** (coordinateCell.y < 0) coordinateCell.y = 0.0;

event.whereX\_Ucell = (**int**) coordinateCell.x;

event.whereY\_Ucell = (**int**) coordinateCell.y;

}

Introduction de maxAge\_Uday dans le genomeAnimalia :/

## 26/05/2017:

Ajout de la molécule 6-methoxybenzoquinone dans la classe rodent gerbil et végétation

Le 6-metoxybenzoalinone (Hima, 2010) effet stimulateur sur la reproduction démontré (Berger et al., 1987 ; Linn, 1991 cités par Shenbrot & Krasnov, 2001).

Vérification de la reproduction suivant la présence ou non de la molécule 6- methoxybenzoquinone

## Note :

La gerbille reste ni prématuré ni sexuel mature ?

Illogique de se percevoir soit même! Ok pour l’instant

Revoir la probabilité de mourir des gerbilles.

Faire sortir les gerbilles des terriers si c’est la nuit ; constat : les gerbilles ne resteront jamais dans un terrier durant la nuit, noté aussi qu’on a des pas de temps d’une heure. Ok (si chouette, elles rentrent et sortent jusqu’à ce qu’il n’y ait plus de chouette

Se reproduire même si la quantité d’énergie est négative ?

Revoir la gestion de l’énergie sur actionGrowOlder de A\_NDS par pas de temps voir TODO

## 31/07/2017 :

Ajout de else pour **if** (**this**.target == **null**) de step\_Utick de A\_Animal

Ajout du paramètre DISPLAY\_MAP sur parametresèscenario\_Cages

## 20/09/2017

Remarque isDayTime et isNightTime ne marche pas forcément avec les gerbilles car une gerbille sort de son terrier vers 19h et se couche au plus tard vers 7h

## 10.04.2018 suppression de addSexToRodent

**public** **void** addSexToRodent(C\_RodentGerbil one\_rodent, String sex) {

**if** ("MF".contains(sex)) {// determine its gender

String rodentName = one\_rodent.gtThisName();

one\_rodent.setThisName(rodentName.substring(0, 1) + sex + rodentName.substring(2));

**int** sexeGene = ***SEX\_GENE\_Y***;

**boolean** isMale = **true**;

**if** (sex.equalsIgnoreCase("F")) {

sexeGene = ***SEX\_GENE\_X***;

isMale = **false**;

}

((C\_GenomeGerbillusNigeriae) one\_rodent.getGenome()).setGonosome(**new** C\_XsomePairSexual(sexeGene));

one\_rodent.setMale(isMale);

}

// else A\_Protocol.event("Protocol Gerbil, gerbil event - gender unknown: " + sex, isError);

}

## 13.04.2018

- hasToDisperse devient isHasToLeaveContainer

- A\_Animal :

suppression de actionForage dans actionWander

modification de actionDisperse

- barnOwl en cours

- rodentGerbil: modification du step

- correction des intervalles de temps journaliers

- correction du protocole gerbille remplacement de contextualizeNewAgentInContainer par contextualizeNewAgentInSpace et suppression de addSexRodent

- Correction des priorités sur les pertes de désir avant la reproduction

- Changement de hasToDisperse par hasToLeaveFullContainer

! sensing = speed\*3

## 26.04.2018 : svn n°1058

- Amélioration de la perception des cellules dans landscape (canInteractWithCell)

- Sélection de la destination la plus proche dans A\_Animal#selectDestination

- Correction de test dispersal dans sethasToLeaveFullContainer

**A FAIRE** : getDistance\_Umeter de gerbil, distance de repast Simphony et computeObjectsDistance\_Umeter de landscape redondants

- Batch gerbil, Ajout int\_Pop\_SIze dans scenario\_parameters, ajout de nestle\_icon

- changement de la visibilité de contextualizeNewThingInSpace dans A\_Protocol

- Modification actionMakeNest à ActionGiveNest sur protocole gerbille

- Suite de mise au point de BarnOwl

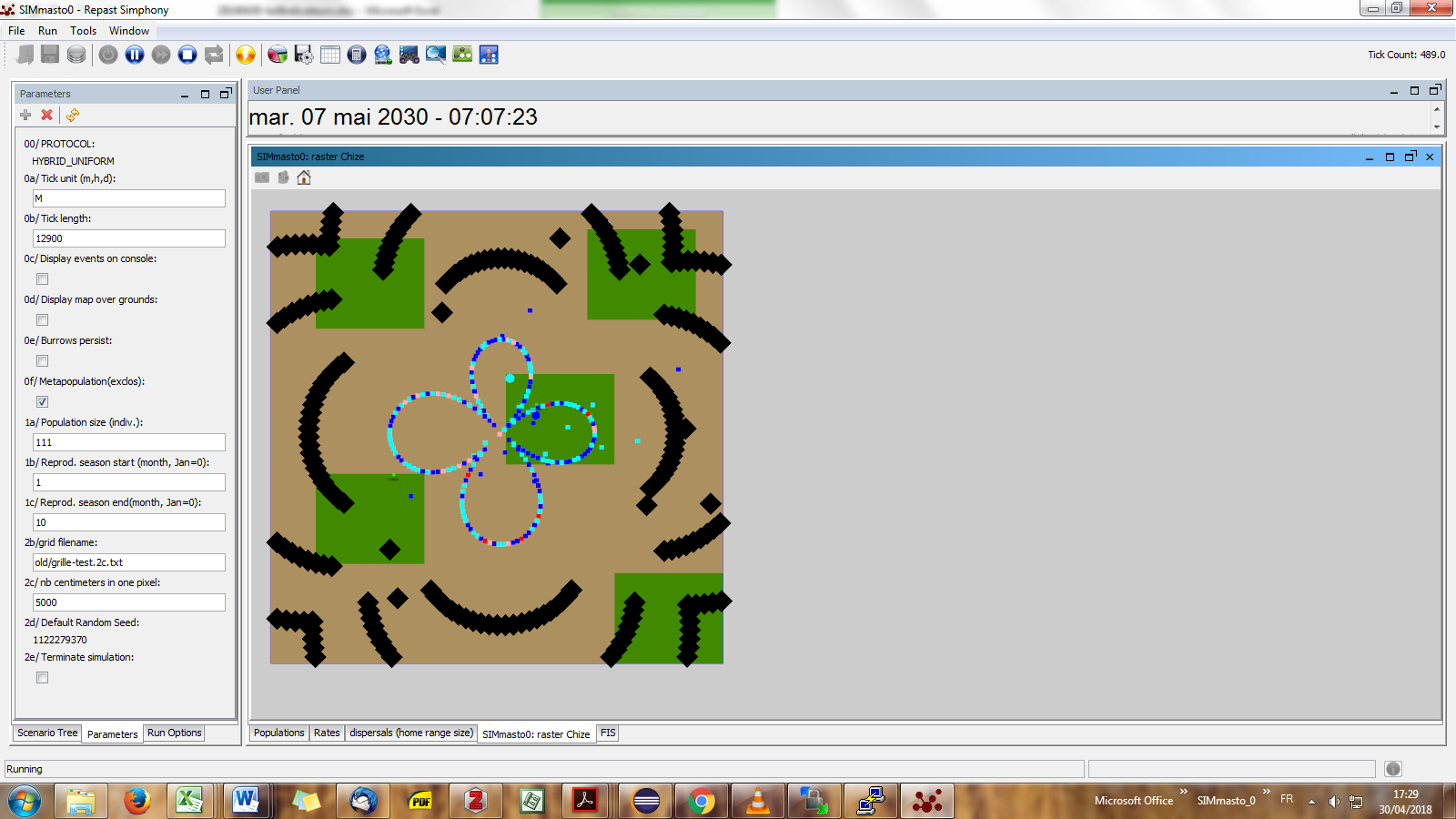
- Ajout de updatePhysiologicStatus dans rodentGerbil (readyToMate different de Amniote)

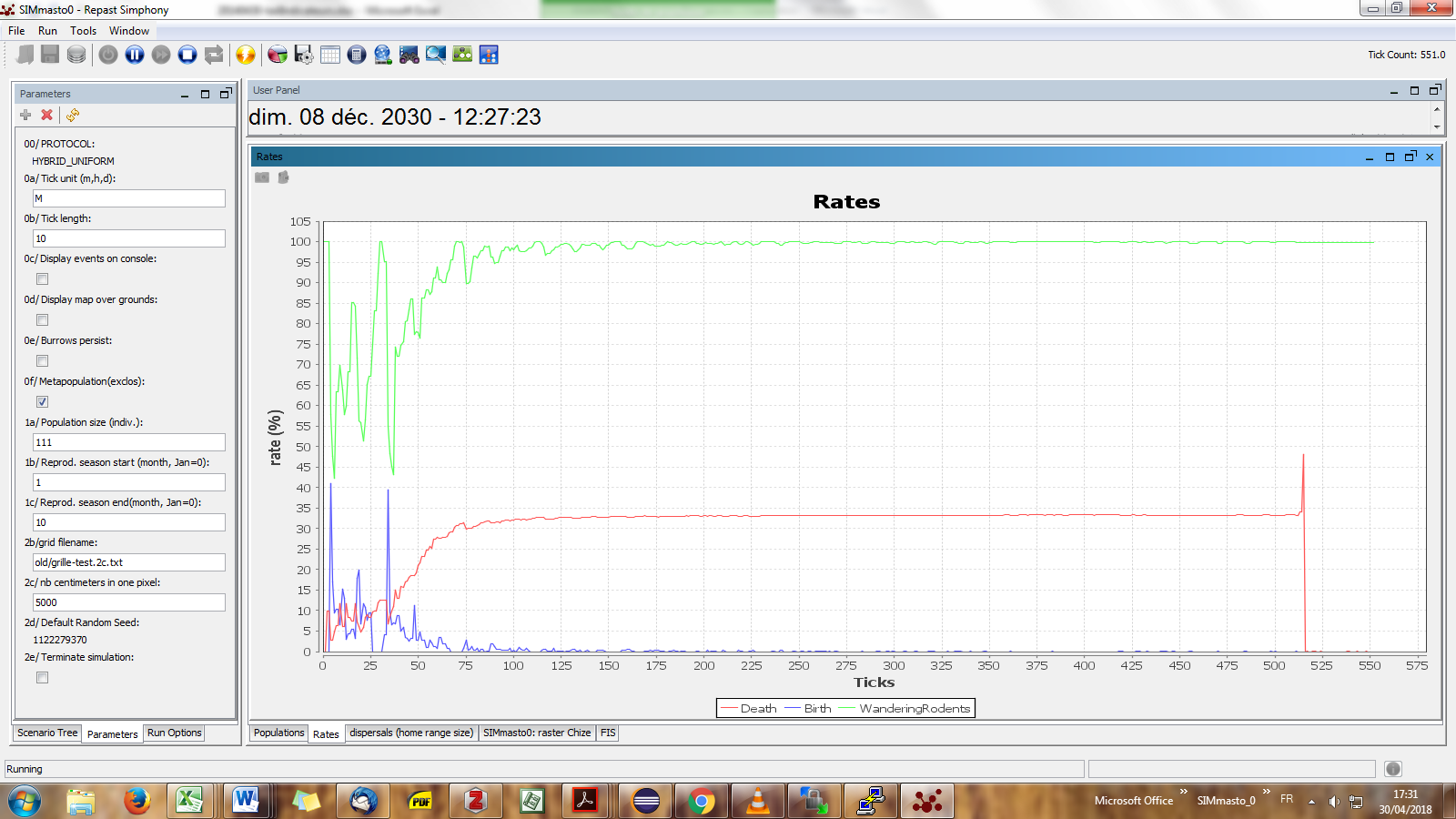
## 27.04.18 modif annulée pour changer la dispersion

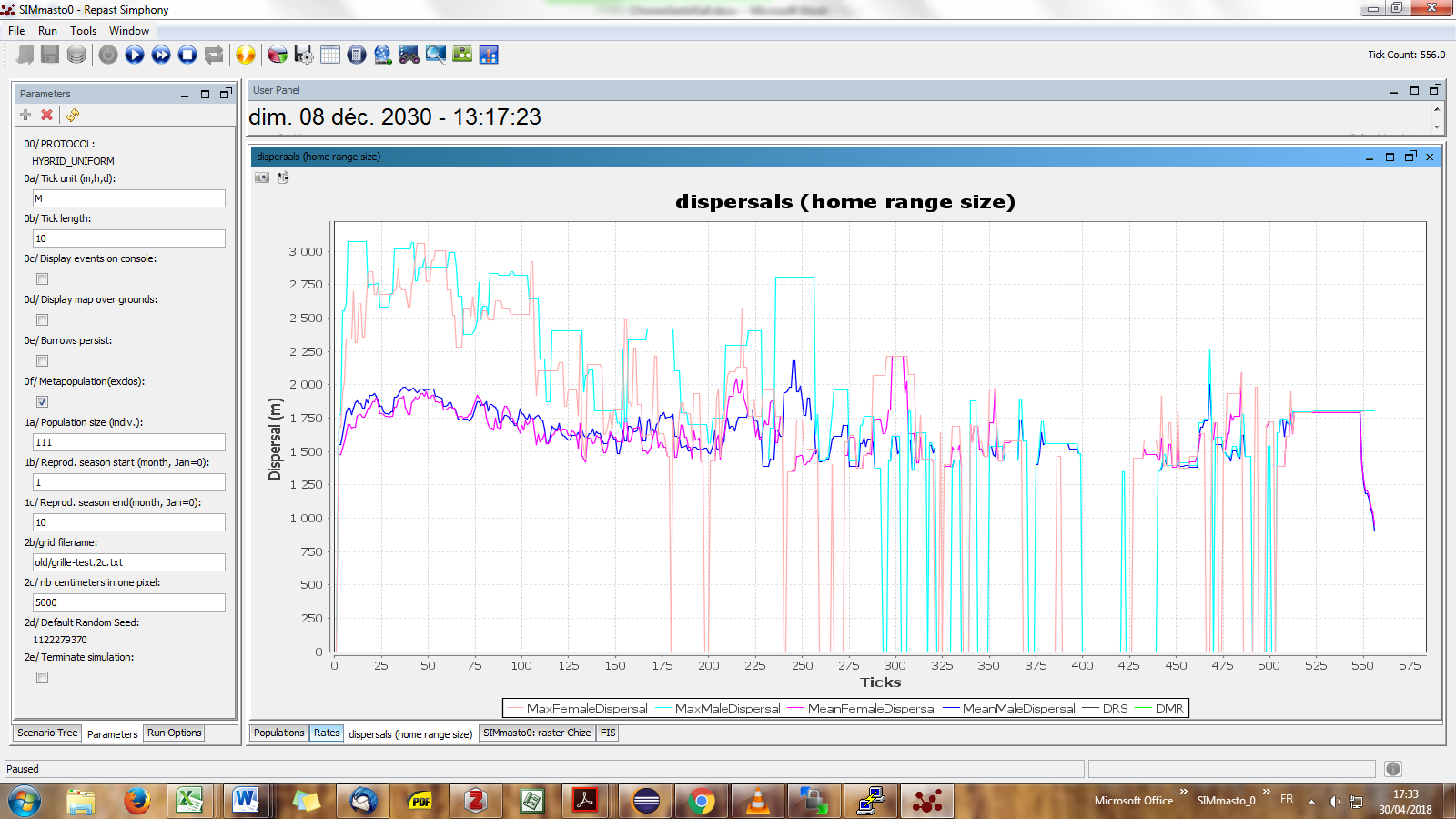
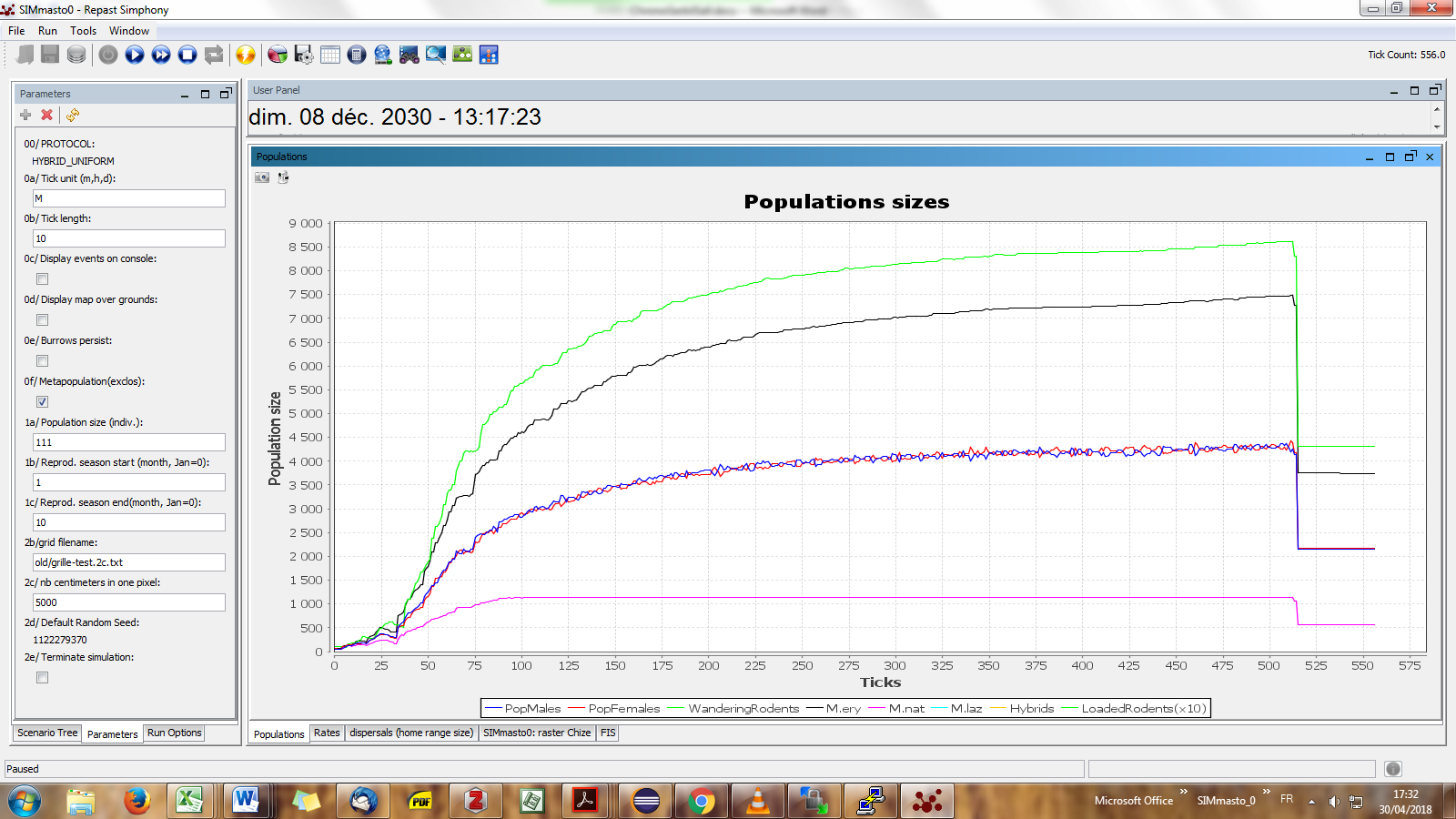
### Dans A\_Animal.deliberation :

**if** (**this**.hasToLeaveFullContainer && **this**.lastContainerLeft != **null**) perceivedThings.remove(**this**.lastContainerLeft);// Do not reenter the container just left

## 30.04.2018 Analyse sensibilité Chize2 fun







### SVN 1061 JUNK commit avec modifications temporaires pour analyses sensibilité SimulTech 2018

## 19.06.2018 : svn n°1071

Commit Temporaire:

1- Ajout des références pour la biomasse végétale et nombre de rongeur maximal contenue dans une cellule.

2- Ajout du choix de l'objet le plus proche dans A\_Animal

3- Mise en place des premieres briques pour une dispersion de l'agent lorsque la zone perçue est pleine.

4- correction de la délibération de la chouette qui retourne dans son nid à l'aube

5- Ajout de checkDanger sur rodent gerbil, correction de chooseShelters

6- Ajout de process target

7- Ajout de actionForage et actionDig

8- Calcul de capacité de charge en rongeur de la cellule suivant sa taille.

## 25.06.2018 : svn n°1072

1- Ajout de canInteractWithThing dans A\_Animal

2- Ajout sur la recherche d'abri à l'aube

3- Ajout de l'étude de cas Dodel2 et simulation des bordures

05.10.2018 : svn n°1082

-Ajout de la classe C\_Human,

-Ajout de l'interface I\_ConstantDodel2,

-Ajout de tous les rasters autour du marché

-Ajout des icônes humains et gestion de l'affichage sur C\_IconSelector

-Ajout du paramètre RasterList\_URL

-Ajout de la lecture de plusieurs raster sur C\_Landscape