














TUTORIEL

Utilisation de l'environnement de simulation

1 – Contenu de l'environnement

Tout d'abord, sur le logiciel Gama, l'environnement de simulation est présenté sous la forme:

- ▼  *includes*
 - >  *images*
 - >  *shapefiles*
 -  *data_camera.csv* (3x1 | no header | delimiter: ',')
- ▼  *models* (22 models)
 - >  *ConstructionSpecies* (4 models)
 - >  *GenericConstructionSpecies* (3 models)
 - >  *GenericVehicleSpecies* (3 models)
 - >  *OtherSpecies* (4 models)
 - >  *Tests* (2 models)
 - >  *VehicleSpecies* (4 models)
 - >  *Main.gaml* (no experiment)
 - >  *trafficModel.experiment* (1 experiment)

Deux principaux répertoires sont représentés. Models est le répertoire où toutes les différentes espèces sont contenues ainsi que le fichier principal (Main.gaml) et les fichiers d'expérience. En revanche, le répertoire includes sert quant à lui à contenir toutes les données extérieures aux espèces comme par exemple les fichiers shapefile des routes, des icônes pour la représentation de certaines espèces...

Comme nous pouvons le voir sur l'image ci-dessus le répertoire models contient lui-même plusieurs sous répertoires. Le répertoire GenericVehicleSpecies contient toutes les espèces génériques permettant la création d'espèces véhicules. Il contient Vehicle.gaml, NonGuidableVehicle.gaml et GuidableVehicle.gaml. Vehicle est l'espèce mère des deux autres espèces, qui quant à elle font la différence entre les véhicules qui pourront être expérimentées par connexion Bluetooth et ceux qui ne pourront pas l'être. Ensuite, le répertoire VehicleSpecies contient les espèces filles de celles présentées juste avant, ce sont les espèces qui seront représentées sur l'interface graphique de la simulation Gama. Nous y trouverons par exemple l'espèce Bike ou encore BluetoothCar qui comme son nom l'indique, correspond aux voitures connectées en Bluetooth pour l'expérimentation.

Concernant les différentes constructions, le schéma est le même, dans le répertoire GenericConstructionSpecies nous avons les espèces mères de tout type de constructions (routes, bâtiments) et dans le répertoire ConstructionSpecies sont contenues les espèces filles.

Voici un schéma représentant les différents liens de parenté entre les espèces déjà présentes dans l'environnement:

Vehicle

- NonGuidableVehicle
 - Bike
 - Train
- GuidableVehicle
 - BluetoothCar
 - BluetoothBus



Construction

- Building
 - Home
 - Administration
- Road
 - Insertion
 - Impasse



Enfin, les deux derniers répertoires présents donnent accès aux tests unitaires réalisés (répertoire Tests) et le répertoire OtherSpecies sert à contenir les espèces qui n'ont aucun lien de parenté avec d'autres espèces.

Sur ce lien: [https://github.com/Shinooow/TrafficModeling/blob/master/Gestion%20de%20Projet/Modeling-Agents%20\(1\).pdf](https://github.com/Shinooow/TrafficModeling/blob/master/Gestion%20de%20Projet/Modeling-Agents%20(1).pdf) vous trouverez le diagramme des agents de l'environnement qui donne de plus amples informations sur les liens de parentés ainsi que les attributs de chaque espèce.

2 – Exemple type de création d'une nouvelle espèce d'agent

Dans cet exemple de création d'une nouvelle espèce d'agent, nous allons créer une espèce de bus (appelée Bus) qui ne sera pas expérimentée dans la simulation réelle, c'est à dire sans connexion Bluetooth. Pour cela nous devons donc créer une espèce qui hérite de NonGuidableVehicle (Note: à chaque utilisation d'une autre espèce, il faut importer le fichier correspondant).

Une fois que le fichier et l'espèce sont créés, il faut mettre à jour tous les attributs hérités que nous souhaitons (voir sur le diagramme des agents de l'environnement) par exemple la vitesse maximale, l'icône qui permet sa représentation... Des actions ou autres fonctions peuvent être ajoutées en fonction du besoin de l'utilisateur.

Maintenant, allons dans le fichier Main.gaml (en pensant à importer le nouveau fichier). Pour créer et ajouter des instances de ce nouvel agent dans la simulation, il faut aller dans le

bloc d'instruction init et utiliser l'instruction create offerte par Gama. Dans le bloc de create, des propriétés peuvent être initialisées. Il faut obligatoirement initialiser le numéro qui identifie l'instance appelé id. Dans notre cas, nos bus étant des véhicules, nous devons incrémenter la variable id_vehicle_created après avoir initialisé la propriété de l'id du bus pour que celle-ci reste correcte et unique pour chaque véhicule. Enfin, il faut impérativement ajouter ces bus dans la liste des véhicules de la simulation pour qu'ils soient affectés par les différents traitements possibles. Voici un exemple de code correspondant à la création des bus.

```
create Bus number: 10 {  
  self.id <- id_vehicle_created;  
  id_vehicle_created <- id_vehicle_created +1;  
  add self to: vehicules;  
}
```

Une fois les bus créés, la dernière étape est de les afficher dans la simulation. Pour cela allons dans le fichier trafficModel.experiment dans la partie output → display map il faut rajouter species Bus aspect: **aspect choisi (voir diagramme d'agents)**. Ce qui dans notre cas donnerait:

```
output {  
  /* Display MAP: affichage des especes */  
  display "map" type: opengl{  
    species Road aspect: base;  
    species Checkpoint aspect: base;  
    species BluetoothCar aspect: with_icon;  
    species Bike aspect: with_icon;  
    species Home aspect: base;  
    species Bus aspect: with_icon;  
  }  
}
```