

DOCUMENTATION

Espèce globale MAIN

Le bloc d'instructions global du fichier Main.gaml contient toutes les données qui sont accessibles à n'importe quel endroit dans le code. Il possède aussi des actions et des réflexes qui sont appliqués à chaque étape de la simulation

1 – Attributs/Propriétés

- date starting_date : date et heure dans l'environnement au commencement de la simulation
- int expChoice : entier correspondant au numéro de l'expérience testée.
- string pathRoad : chemin du shapefile permettant la création des routes dans la simulation
- string zPathCheck : chemin du shapefile permettant la création des checkpoints dans la simulation
- float step : durée d'une étape de la simulation par rapport à la date de l'environnement
- file roads_shapefile : fichier shapefile des routes de la simulation
- file checkpoints_shapefile : fichier shapefile des checkpoints de la simulation
- file railway_shapefile : fichier shapefile des voies ferrées de la simulation
- geometry shape : forme du cadrage de la simulation (se crée depuis le fichier shapefile des routes)
- graph roadGraph : graphe des routes permettant les déplacements des véhicules
- int cycle_time_checkpoint : durée d'un cycle entre deux changements de positions des checkpoints
- vecteur_base : vecteur {0, -1, 0} permettant les calculs des directions et des angles de rotation pour l'affichage des icônes des véhicules.
- seuil_vitesse_min : en raison de la formule du freinage vue au-dessus, un seuil doit être initialisé pour que le véhicule s'arrête en cas de freinage sous ce seuil.
- nb_crashed_cars : nombre de véhicules accidentés (affichage d'un graphique)
- nb_step : nombre d'étapes effectuées dans la simulation à l'instant t
- mise_a_echelle : coefficient permettant de mettre à l'échelle les données récupérées depuis la caméra pour les utiliser sur la simulation Gama
- file camera_data : fichier csv contenant les données fournies par la caméra
- matrix camera_data_matrice : fichier csv transformé en matrice pour permettre sa lecture

- list<Vehicles> vehicules : collection contenant TOUS les véhicules de la simulation
- list<Construction> construction : collection contenant TOUTES les constructions de la simulation
- list<float> row_list : liste correspondant à la première ligne de la matrice camera_data_matrice contenant toutes les données nécessaires de la caméra
- int idGuidableVehicleCreated : variable permettant d'attribuer l'id d'un GuidableVehicle à sa création
- int idNonGuidableVehicleCreated : variable permettant d'attribuer l'id d'un NonGuidableVehicle à sa création
- int idConstructionCreated : variable permettant d'attribuer l'id d'une construction à sa création
- int idAgentCreated : variable permettant d'attribuer l'id d'un agent (humain) à sa création
- int idRuleCreated : variable permettant d'attribuer l'id d'une règle à sa création
- int idSignalisationCreated : variable permettant d'attribuer l'id d'une signalisation à sa création
- float seuilBrakelfCollision : lors du calcul pour savoir si deux véhicules vont se rentrer dedans (pour prévoir un freinage), cette constante est utilisée pour indiquer la distance maximale entre deux véhicules pour qu'ils freinent. Si deux véhicules sont dans deux directions opposées et qu'ils vont se rentrer dedans, ils commenceront à freiner une fois que la distance entre eux sera en dessous du seuil.

2 – Constructeurs

Plusieurs constructeurs sont disponibles à l'installation de l'environnement, l'utilisateur peut choisir d'en ajouter, supprimer ou modifier. **ATTENTION** : les constructeurs ne retournent pas l'objet créé. Voici une liste des constructeurs :

- consBluetoothCar()
- consBike(float vitesse)
- consRoads(string shapefilePath)
- consRuleMessage(string content)
- consRuleOneway(Checkpoint directionTarget)
- consHome(point position, float dimensionLongueur, float dimensionLargeur)
- consAdmin(point position, float dimensionLongueur, float dimensionLargeur)
- consBus(float vitesse)
- consBusWithStartAndTarget(float vitesse, point start, Checkpoint checkpoint)
- consCheckpoints(string shapefilePath)
- consCheckpoint(point position, Construction construction)
- consAgent(point position, Administration admin, Home home)
- consTrain(point position, Railway railway)

- consRailway(point position, float dimensionLongueur, float dimensionLargeur)
(peut aussi être fait depuis un fichier shapefile)

3 – Actions/Fonctions/Reflexes

String getRoadPathFile

Retourne le chemin du fichier shapefile correspondant aux routes en fonction du numéro d'expérience choisi expChoice.

String getCheckpointPathFile

Retourne le chemin du fichier shapefile correspondant aux checkpoints en fonction du numéro d'expérience choisi expChoice.

Action initBasicExp

Action qui crée les agents de la simulation numéro 0 (expérience basique) dont des bus, des vélos et des voitures connectées en Bluetooth

Action initBrakelfCollisionExp

Action qui crée les agents de la simulation numéro 1 (expérience de test pour l'algorithme de détection et prévention de collision entre véhicules)

Action initInsertionExp

Action qui crée les agents de la simulation numéro 2 (expérience de test pour les algorithmes concernant l'insertion dans une voie d'un véhicule)

Action initRondPointExp

Action qui crée les agents de la simulation numéro 3 (expérience de test pour les algorithmes concernant les ronds-points et les différentes actions possibles dans un rond-point pour un véhicule)

Action initUPSCampusExp

Action qui crée les agents de la simulation numéro 4 (expérience sur le campus de l'université Paul Sabatier)

Reflex one_step

A chaque étape de la simulation, le compteur d'étapes est incrémenté et les données de la caméra sont mises à jour pour permettre une modification sur la simulation

Reflex stop_simulation

S'effectue lorsque nous n'avons plus de données venant de la caméra (50 lignes lues du fichier csv correspondant). Alors tous les véhicules connectés en Bluetooth se déconnectent et la simulation s'arrête.

Contact : maxence.demougeot1@gmail.com