Rapport de stage

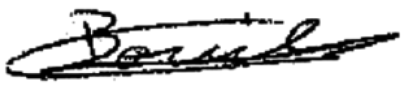
Rapport intermédiaire

Contribution au développement d'un outil de simulation pour l'aide à l'animation de formation à la gestion de crise : création de comportements

Réalisé par :
ITALIANO Lorenzo

Vu, le 18/05/21

Sous la supervision de :
BOUILLET Philippe



Année universitaire 2020-2021

Sommaire :

Table des Figures :

Glossaire

Crise : Désigne un événement qui survient de manière inattendue et déstabilise une organisation. C'est une situation difficile dans laquelle il y a des enjeux à protéger (vies, infrastructures ou économie par exemple).

Cellule de crise : Désigne le melting pot de personnes compétentes et d'élus qui se regroupent tous pour pouvoir gérer une crise le plus efficacement possible.

Gestion de crise : Désigne les modes d'organisation, techniques et moyens qui permettent à une cellule de crise de se préparer à faire face à une crise et d'en tirer des leçons afin d'améliorer la gestion pour être mieux préparé dans le futur.

Système multi-agents : Un système multi-agents est un programme dans lequel tout est modélisé sous forme d'agents qui interagissent entre eux pour créer une simulation.

Gama platform : C'est un environnement de développement open-source basé sur un système multi-agents.

Action (dans Gama) : Désigne un comportement d'une espèce. Cela peut être comparé à une fonction en programmation orientée objet.

Reflex : Désigne une action récurrente qui peut soit se produire chaque pas de simulation (par exemple pour un compteur) ou lorsqu'une condition est valide.

Espèce : Désigne un type d'agent, c'est exactement ce que représente une classe en programmation orientée objet. Il est possible de faire des espèces abstraites.

Sous-espèce : Désigne une espèce qui hérite d'une autre, comme une classe java pourrait hériter d'une autre (par exemple une des sous-espèce de l'espèce ‘‘animaux’’ est “singe”).

Experiment : représente l'affichage de la simulation que l'on va lancer, permet de définir ce que l'on affiche à l'écran, ce qui peut être des paramètres changeables et les variables affichées.

Global : Partie où l'on déclare les variables et les fonctions globales à un modèle.

init : Sous partie dans laquelle on initialise les valeurs de certaines ou toutes les variables d'un élément (global ou espèce).

Monitor : Permet d'avoir la valeur d'une variable affichée dynamiquement lorsque la simulation s'exécute.

Parameter : Permet de mettre une variable en paramètre pour que l'on puisse changer sa valeur avant de lancer la simulation ou pendant que la simulation s'exécute.

Model : Un modèle désigne un fichier .GAML. Un modèle contient la déclaration d'une ou plusieurs espèces ainsi que des variables et comportements globaux ou dépendants d'une ou plusieurs espèces.

Skills : Désigne des capacités programmés dans Gama que nous pouvons attribuer aux espèces. Par exemple, le skill move qui permet aux agents de se déplacer.

Introduction

Lorsqu'une crise se présente, il est nécessaire de la gérer et de la régler le plus rapidement et efficacement possible pour pouvoir éviter le maximum de dégâts générés par la crise. Il y a plusieurs piliers de la gestion de crise, on peut notamment retrouver 3 pans importants :

-Diagnostic, action et décision
-Organisation
-Communication

Lorsqu'une crise est assez conséquente (incendie, incident industriel ou autre), une cellule de crise est mise en place pour pouvoir tenter de gérer au mieux cette crise et de faire en sorte de protéger la population. Selon la crise et la taille de la circonscription dans laquelle se passe l'incident, la cellule de crise s'appuie sur des principaux décideurs (maire, adjoints, etc ...), des experts au regard de la situation (pompiers, police, militaires, scientifiques ou autre selon la situation et la nature de l'incident), des professionnels de la communication pour tenir informé les gens et des juristes, assureurs. Il est donc important que ces personnes, qui en principe n'ont jamais ou très peu travaillé ensemble en même temps, puissent travailler dans les meilleures conditions et gérer au mieux la situation, puisque dans la plupart des cas, des vies sont en jeu.

La problématique qui se pose ici est de trouver une façon de former les personnes compétentes à bien réagir en cas de crise et en cas de gestion de crise, pour que la cellule de crise puisse agir le mieux possible et prendre les meilleures décisions afin que la gestion de l'incident soit la meilleure possible.

Une des solutions existantes pour pouvoir former les personnes concernées est la simulation de crise. C'est à dire que on va dérouler un scénario mis en place par des animateurs, dans lequel se produit une crise, effectuer des rejeux de certains moments lorsque la cellule de crise prend une trop mauvaise décision, puis faire un retour d'expérience à la fin de l'exercice pour expliquer aux apprenants les points faibles et forts de leur gestion de crise afin qu'ils puissent s'améliorer.
Ce projet s'inscrit précisément dans le rejeu de certains moments de la situation. Pour pouvoir gérer au mieux le scénario de manière plus simple, une simulation en programmation multi-agents est une bonne solution puisque les agents ne réagiront pas

toujours de la même façon ce qui rend l'exercice plus réaliste et plus vivant. Cette simulation permettrait donc de gérer au mieux le scénario pour les animateurs qui pourraient influer sur la simulation selon les décisions des apprenants afin de pouvoir leur donner les informations sur la situation en fonction de la simulation et de la réaction des agents. Il sera ici question d'améliorer une simulation de gestion de crise déjà existante au sein du LSR qui a été créée l'année dernière par une stagiaire. L'objectif de ce stage est d'améliorer et ajouter de nouvelles fonctionnalités et de nouveaux agents qui n'existent pas encore dans la simulation existante.

sigle ?

Nous allons donc présenter ce stage et ce projet plus en détail dans différentes parties et sous parties. Nous verrons tout d'abord une présentation de l'entreprise avec les 2 entités qui supervisent le laboratoire. Puis, nous verrons l'analyse avec la présentation détaillée du projet, l'analyse détaillée de l'existant (la simulation développée précédemment) ainsi que la présentation et explication du cahier des charges. Finalement, nous verrons les spécifications techniques avec une brève présentation de ce qu'est la programmation multi-agents et comment elle fonctionne ainsi que la présentation des principales technologies utilisées pour le projet (Qgis et Gama Platform).

I - Présentation de l'entreprise

J'ai réalisé mon stage au sein du Laboratoire des Sciences des Risques (LSR). Cette première partie sera consacrée à la présentation de ce laboratoire de recherche. Cependant, ce laboratoire dépend de deux autres entités qui sont respectivement L'Institut Mines-Télécom et IMT Mines-Alès. Nous allons donc dans un premier temps introduire les 2 entités qui supervisent Le LSR puis dans une troisième sous partie, nous présenterons le LSR.

1.1 - L'Institut Mines-Télécom

L'institut Mines-Télécom (IMT), grand établissement au sens du code de l'éducation, est un établissement public scientifique, culturel et professionnel (EPSCP) placé sous la tutelle principale des ministres chargés de l'industrie et du numérique. Premier groupe d'écoles d'ingénieurs en France, il regroupe 11 écoles d'ingénieurs publiques réparties sur le territoire national, qui forment 13 500 ingénieurs et docteurs. L'IMT emploie 4500 personnes et dispose d'un budget annuel de 400 M€. L'IMT comporte 2 instituts Carnot, 35 chaires industrielles et produit annuellement 2100 publications de rang A, 60 brevets et réalise 110M€ de recherche contractuelle.

1.2 - IMT Mines Alès

Créée il y a 178 ans, IMT Mines Alès compte à ce jour 1200 élèves (dont 200 étrangers) et 350 personnels. Elle possède deux campus à Alès (le site de Clavières et le campus technologique Louis Leprince Ringuet) et est également implantée à Pau. L'école forme des ingénieurs généralistes, des ingénieurs de spécialités (par apprentissage), des doctorants et des élèves de masters ou mastères spécialisés. Elle accueille plus de 500 stagiaires en formation continue professionnelle. L'école dispose de 3 centres de recherche de haut niveau scientifique et technologique, qui œuvrent dans les domaines des matériaux et du génie civil (C2MA), de l'environnement et des risques (CREER), de l'intelligence artificielle et du génie industriel et numérique (CERIS). Ces entités regroupent environ 80 enseignants-chercheurs permanents (dont 40 Habiliation à Diriger les Recherches), 20 personnels techniques et 10 personnels administratifs de soutien à la recherche, 80 doctorants et post-doctorants, qui produisent chaque année 90 publications de rang A et 3M€ de contrats de recherche, dont 1M€ de contrats directs avec les entreprises. IMT Mines Alès est accréditée à délivrer le diplôme de docteur dans 4 écoles doctorales. Elle dispose de 12 plateformes

technologiques et compte 1600 entreprises partenaires. L'école fut la première à créer un incubateur d'entreprises en 1984 (200 entreprises créées à ce jour, 1000 emplois).

IMT Mines Alès a noué des partenariats structurants avec le CNRS et les universités de Montpellier, de Nîmes et de Pau. Les centres de l'école ont en particulier développé des collaborations scientifiques solides avec les unités de recherche HSM, LMGC, IPREM, EUROMOV et CHROME.

1.3 - Le Laboratoire des Sciences des Risques

Le Laboratoire des Sciences des Risques (LSR) est une unité de recherche d'IMT Mines Alès destinée au développement des travaux de recherche centrés sur la gestion des risques et plus particulièrement sur l'amélioration de la sécurité, la sûreté, et le bien-être des populations et des générations futures face aux risques technologiques, chroniques, naturels et sanitaires. Ce laboratoire de recherche est constitué de deux équipes d'IMT Mines Alès, adossées aux centres d'enseignement et de recherche CREER et CERIS.

Ainsi, les centres CREER (Centre de Recherche et d'Enseignement en Environnement et en Risques) et CERIS (Centre d'Enseignement et de Recherche en Informatique et Systèmes) collaborent au sein du LSR au travers de leurs équipes EUREQUA (EtUdes des Risques et la QUalité de l'Air) et ISOAR (Ingénierie des Systèmes et des Organisations pour les Activités à Risques).

· L'équipe EUREQUA (12 enseignants-chercheurs dont 7 HDR, 3 Ingénieurs de recherche, 3 techniciens, 10 doctorants) développe une recherche ciblée sur la gestion des risques majeurs, les pollutions par les COV (Composé Organique Volatil), les nuisances et gênes liées aux odeurs. Les champs d'application concernent les risques industriels, naturels et les risques chroniques liés aux rejets industriels ou les ambiances confinées (air intérieur, ambiance de travail).

L'équipe ISOAR (Ingénierie des Systèmes et des Organisations pour les activités à Risques) comporte 9 enseignants-chercheurs et 9 Doctorants. Les travaux réalisés dans l'équipe ISOAR s'intéressent à l'aide à apporter à un collectif d'acteurs pluridisciplinaires pour mieux maîtriser le risque lorsqu'ils mènent des activités en lien avec des systèmes réputés complexes, en particulier, des systèmes de systèmes et des organisations. Les activités visées ici consistent à concevoir, optimiser, vérifier et valider, évaluer ces systèmes puis à décider et justifier ces décisions avant toute réalisation. Elles portent ensuite sur la réalisation et le déploiement de ces systèmes, la

formation des opérateurs et gestionnaires, le pilotage et la gestion de l'ensemble en toute situation. Elles couvrent enfin le maintien de ces systèmes en conditions opérationnelles en phase d'exploitation avant leur démantèlement en fin de vie.

Le LSR est structuré scientifiquement autour d'une double approche scientifique en croisant les 4 thèmes de recherche suivants :

-Caractérisation et réduction des aléas ;
-Evaluation de la vulnérabilité et de la résilience des enjeux ;
-Ingénierie des systèmes complexes face aux risques ;
-Gestion de crise.

Ces thèmes de recherche constituent des questions scientifiques transverses et fédératrices appliquées à 4 « champs applicatifs », qui constituent des types de risques pris pour objets d'études (les risques technologiques, chroniques, naturels et les risques sanitaires exceptionnels).

II - Analyse

Dans cette partie, nous allons faire l'analyse du sujet de stage. Il sera donc question d'analyser le sujet et les missions du stage. Pour cela, nous séparerons cette partie en 3 sous-parties. Nous verrons tout d'abord en quoi consiste le sujet et dans quel contexte il s'inscrit, puis nous verrons ensuite l'analyse de l'existant avec la première version de la solution. Finalement, nous verrons le cahier des charges en l'analysant et en le détaillant.

2.1 - Présentation du sujet et analyse du contexte

Le Laboratoire des Sciences des Risques (LSR) réalise des exercices de formations à la gestion de crise, visant à améliorer les compétences des apprenants en matière de gestion de crise (exemple vidéo d'un exercice de 2016 https://www.youtube.com/watch?v=OcaAg_zaSdk).
Au sein du LSR, des travaux de recherches actuels portent sur le rejeu d'une partie ou de l'entièreté d'un scénario d'exercice. C'est-à-dire revenir en arrière avant un événement déclencheur qui a été mis en place soit trop tard, soit d'une mauvaise manière ou bien même car une mauvaise décision a été prise et à des conséquences négative soit sur la population soit sur la cellule de crise (évacuation effectuée trop tardivement, mauvaise communication d'une information, etc…). Ces rejeux visent à montrer aux apprenants quelles décisions auraient pu être prises pour pouvoir au mieux gérer la crise et protéger la population.
Dans le cadre de ces recherches, un outil de simulation a été développé grâce à un système multi-agents open source Gama Platform (https://gama-platform.github.io/). Cette simulation permet de modéliser tous les acteurs présents dans les scénarios de crise (civils, policiers, pompiers, véhicules, etc...). Le système multi-agents permet de modéliser tous les acteurs du scénario, en leur ajoutant des caractéristiques et des comportements propres ainsi que des interactions entre eux.
Les principaux objectifs de la simulation sont de suivre le scénario établi par les animateurs et évaluer les conséquences potentielles des décisions prises par les apprenants dans la cellule de crise afin de déterminer lorsqu'il faut procéder au rejeu.

Dans la première version du projet qui a été réalisé par une stagiaire l'année dernière, un scénario précis a été sélectionné pour être implémenté dans la simulation. Le scénario est issu d'une crise réelle qui a eu lieu dans la ville de Carnoux-En-Provence, cet incident était un feu de forêt qui a eu lieu en 2017.

L’objectif ultime du projet est de pouvoir développer une solution adaptable à n’importe quelle ville afin de pouvoir simuler n’importe quel type de crise, que ce soit un incendie naturel (incendie, ouragan, tsunami ou autre) ou bien industriel (explosion, déversement de produits chimiques dans l’air, l’eau ou autre).

Les principaux objectifs de mon stage sont d’ajouter de nouvelles fonctionnalités à la simulation existante. C’est à dire d’ajouter de nouveaux agents qui n’existent pas encore dans la simulation (Feu, Vent, Fumée, Sapeurs-Pompiers, Bus par exemple). De plus, il sera question d’améliorer certains agents, dans leurs comportements et dans leurs interactions avec tous les autres agents. Également, certaines fonctionnalités générales de la simulation sont à revoir ou à améliorer. Par exemple, le système de déplacements, notamment en véhicule, doit être revu pour être plus réaliste (collisions, embouteillages et feux de circulation).

2.2 - Analyse de l’existant

L'école des mines d’Alès n’est pas la seule institution à faire des recherches sur la gestion de crise. En effet, d’autres institutions travaillent sur le sujet, et font des simulations de crise pour former leurs apprenants. Parmi ces institutions, on peut retrouver notamment en France l’École nationale supérieure des mines de Nancy, l'Université Lille I, l’INSEEC, l'Institut National des Sciences Appliquées (INSA) de Rouen et en science politique à l'Université Lyon 3.

Au LSR, il existe une version 1 du projet de simulation multi-agents qui a été développée par une stagiaire l’année dernière. Cette première version a été nommée CÉSAR (Code d’Étude Simulé pour les Agents à Rejouer). Nous allons donc partir de cette version existante afin de l’améliorer et la compléter. Cette simulation prend place dans la ville de Carnoux-En-Provence (13), afin de dérouler un scénario qui s’est réellement passé en 2017. En effet, le 19 Août 2017, un incendie s’est déclaré dans une forêt accolée à Carnoux-En-Provence et une gestion de crise s’est donc mise en place, c’est donc cet événement qui est reproduit dans CESAR. Cette simulation a été faite sur l’environnement open source Gama Platform et les informations géographiques ont été extraites et traitées avec le logiciel Qgis, ce sont donc les choix technologiques qui ont été faits et nous allons poursuivre avec cette combinaison d’outils.

Dans cette première version, on peut retrouver pas mal d’éléments qui ont déjà été intégrés. Tout d’abord la simulation intègre les informations géographiques de Carnoux-En-Provence (bâtiments, système routier et forêts notamment). On retrouve aussi une panoplie d’agents existants: habitants et touristes, police, employés de mairie et utilisateur. Le système de déplacements des agents sur la route a été fait, même s’il faudra le revoir car la notion de véhicule n’existe pas dans cette version. On peut de

plus agir sur la simulation en utilisant des commandes utilisateur. On peut par exemple envoyer des agents de mairie ouvrir des centres d'hébergement d'urgence, envoyer la police confiner un quartier ou leur demander d'évacuer de plusieurs manières (porte à porte ou mégaphone).

Les plus gros problèmes de cette première version sont l'architecture et les commentaires dans le code. En effet, CÉSAR V1 est un seul fichier d'environ 1800 lignes de code, ce qui rend la reprise du code et la maintenabilité de ce code assez complexe, il faudra donc ré-architecturer le projet en tentant de séparer les éléments qui le peuvent. De plus, dans ce code, il y a très peu de commentaires. En effet, une documentation très détaillée est fournie avec le code, cependant il est assez compliqué de naviguer entre le code et la documentation lorsqu'on ne connaît pas encore bien le code, il sera donc nécessaire de bien commenter le code à venir ainsi que commenter le code déjà existant.

2.3 - Cahier des charges

Mon cahier des charges a été rédigé par l'utilisateur final de l'application (cf. Annexe 1). Ce cahier des charges comporte plusieurs parties dans lesquelles il y a plusieurs tâches à faire, de plus il est indiqué à côté de chaque tâche sa priorité (très faible, faible, moyenne, forte ou priorité maximale). Cela permet de pouvoir trier les tâches à faire et de prioriser les plus importantes en premier ou tout du moins dans les premières. Ce cahier des charges se compose donc de 5 parties: Mise à jour des agents existants, Mise à jour des consignes existantes, Ajouts de nouveaux agents, Ajout de nouvelles « consignes » et Autres. Nous allons voir de manière plus détaillée ce que contiennent ces parties et voir leur niveau de priorité.

point virgule entre les éléments d'une énumération.

Tout d'abord, la partie la plus importante de ce cahier des charges, la mise à jour des agents existants. En effet, dans cette partie il n'y a que deux tâches, cependant ce sont les tâches les plus prioritaires et les plus urgentes. Premièrement, il faut revoir le fonctionnement des déplacements pour corriger ce qui existe, ajouter les feux de circulation et la notion de véhicule (pour pouvoir gérer les collisions et embouteillages). Pour cela une solution intégrée à Gama existe, il faudra cependant programmer un bon nombre de choses et gérer certains problèmes pour pouvoir mettre cela en place. Deuxièmement, il faudra implémenter un comportement de routine pour tous les agents humains, pour rendre vivant et réaliste la simulation. En effet, actuellement, si on ne demande pas aux civils d'évacuer ou de se confiner, ils restent chez eux sans rien faire de spécial, ce qui n'est évidemment pas le cas dans la réalité.

Ensuite, il faudra mettre à jour les consignes existantes. Cependant ces tâches sont en priorité faible, il ne faut donc pas commencer par cela, ni passer énormément de temps dessus s'il y a des tâches plus importantes à faire avant. Les tâches de cette partie consistent à mettre à jour des consignes déjà existantes dans CÉSAR telles que la consigne d'évacuation, de confinement et les consignes d'hébergement. Il sera aussi question de rectifier la commande qui indique aux policiers d'envoyer les consignes à suivre. En effet, dans cette commande, on nous propose en entrée d'indiquer sous forme de String la manière dont on fait passer l'information (porte à porte, haut-parleur, internet ou radio). Il faudrait donc faire en sorte que l'utilisateur soit au courant des possibilités qu'il a, soit par une indication ou bien par une liste à choix par exemple.

Une des tâches les plus importantes pour pouvoir améliorer la simulation est l'ajout de nouveaux agents. Cela est très important pour enrichir le scénario. Certains éléments devront être totalement créés, alors que pour d'autres, il suffira d'ajouter quelques lignes de code. En effet, par exemple, pour ajouter des habitants dans tous les quartiers ou bien pour ajouter tous les bâtiments d'hébergement, il n'y aura pas beaucoup de développement à faire, puisque l'agent habitant est existant et un centre d'hébergement est déjà disponible. D'une part, il y a des agents qu'il faut créer de toutes pièces, cette partie sera plus chronophage. Parmi les agents à développer on trouve le feu, les pompiers et le matériel de pompier, le vent, la fumée, les gendarmes, le service technique de la mairie, des outils, le service logistique, les bus, les services médicaux d'urgence, des agents du système ferroviaire, le personnel de l'éducation nationale.

Il est aussi question d'ajout de nouvelles consignes que l'on pourrait donner en tant qu'utilisateur pour agir sur la simulation. Un exemple de consigne est la consigne qui indique à un ou plusieurs policiers d'aller évacuer un certain quartier. Dans les consignes à ajouter il y a des consignes permettant d'influer sur les segments de route (vitesse, types de véhicules autorisés, sens de circulation), des consignes permettant de jouer sur le positionnement des pompiers et de pouvoir régler leurs effets sur le feu, une consigne permettant de livrer du matériel et une consigne PPMS incendie activé/désactivé pour chaque bâtiment.

Finalement, il reste les autres tâches à faire qui sont deux tâches d'ergonomie. Ces deux tâches sont assez importantes, puisqu'elles concernent l'ergonomie et qu'elles sont notées comme priorité forte. Ces deux tâches sont la redéfinition du code couleur des agents dans la simulation et la redéfinition des données paramétrables en début de la simulation. Cependant, cela va se faire dans le temps, il n'est pas nécessaire de faire tout d'un coup, mais de bien penser à ces deux éléments lorsqu'on intègre un nouvel agent ou un nouveau paramètre important.

III - Spécifications techniques

Les spécifications techniques sont pour la plupart imposées par la première version de la simulation programmée précédemment au sein de l'équipe. Dans ces spécifications, nous retrouvons le type de programmation utilisé ainsi que les technologies utilisées afin de mettre en place ce développement. Nous traiterons donc ces deux sujets dans cette partie. Nous verrons tout d'abord la programmation multi-agents en explicitant ses principaux principes ainsi que ces avantages dans notre contexte puis nous verrons le "logiciel" que nous utiliserons pour mettre en place cette programmation multi-agents (Gama Platform) et nous verrons ses principes, son vocabulaire et ses spécificités.

3.1 - Programmation multi-agents

La programmation multi-agents peut se comparer à de la programmation orientée objet, mais serait plutôt décrite par de la programmation orientée agent. C'est-à-dire que la programmation multi-agents est organisée autour d'agents que l'on pourrait comparer à une classe en java par exemple. Ces agents peuvent représenter une infinité de choses diverses telle que des bâtiments, des humains, des animaux, des routes ou bien ce que l'on veut. L'utilité de ce paradigme de programmation est de pouvoir créer des agents qui interagissent entre eux dans le cadre d'une simulation, ce qui fait que chaque simulation sera différente puisqu'elle dépend des interactions avec les agents qui ne sont jamais les mêmes à chaque fois que l'on lance la simulation. Il est à noter que les agents sont autonomes, il suffit de leur implémenter des comportements et des réflexes pour qu'ils soient autonomes. Cela peut nous aider à simuler des événements réels comme un mouvement de foule par exemple, puisque deux mouvements de foule, même dans des situations exactement similaires, ne seront pas identiques, du moins c'est très très peu probable. Il est cohérent d'utiliser ce paradigme de programmation dans notre situation puisque, comme nous l'avons vu plus tôt, le but ici est de simuler des actions humaines lors d'une situation de crise. Or, il serait très difficile voire impossible de faire ce genre de simulation avec un autre paradigme de programmation puisqu'il faut que les simulations dépendent de tout ce qu'il se passe et des interactions avec les autres agents. Il serait possible de recréer ce genre de simulation avec d'autres moyens tels que la programmation orientée objet, mais on aurait soit une simulation qui ne serait pas assez fidèle à la réalité, soit on se retrouverait à reprogrammer le principe d'un système multi-agents puisqu'il faudrait définir comment les personnes interagissent entre elles et ce que cela provoque notamment. C'est donc pour cela que la création de cette simulation passe

nécessairement par un système multi-agents, et c'est pour cela que l'utilisation d'un tel système a été admise pour la création de la version 1.

3.2 - Qgis

Un des logiciels utilisés dans ce projet et qui est indispensable au réalisme de la simulation est Qgis. Qgis est un SIG (Système d'information géographique). C'est un logiciel avec lequel on peut extraire ou manipuler des données cartographiques. C'est-à-dire que l'on a une carte avec par exemple une couche de bâtiments, la couche possède l'information des formes, mais également des données, on peut par exemple avoir le type de bâtiment, s'il est industriel, résidentiel ou autre. Ces fichiers qui contiennent ces informations sont des .shapes (abrégée .shp). Ce type de fichier est très important pour notre simulation car il permet d'importer des informations géographiques réelles très rapidement et simplement. En effet, grâce à cette source d'information nous n'avons pas à refaire la ville à la main, il suffit d'extraire les données open-source par exemple grâce à OpenStreetMap. Nous pouvons aussi nous procurer des .shp grâce à des sites gouvernementaux tels que le site IGN pour les données concernant la France. Il nous suffira ensuite de charger ses fichiers dans la simulation puis de les utiliser pour créer les agents correspondants (agent route issue des données géographiques de la route par exemple).

3.3 - Gama Platform

Le choix de la programmation multi-agents a été justifié précédemment (cf. partie 3.1), il fallait donc choisir un support sur lequel nous pouvons développer en système multi-agents. Ce choix n'a pas été fait lors de mon stage, en effet, ce choix a été fait antérieurement lors du premier stage de l'année dernière. Le choix s'est porté sur Gama Platform, notamment car c'est une bonne solution et qu'elle est open source et gratuite, ce qui signifie qu'elle est assez malléable et que l'on peut aller modifier des choses dans le code source, créer des librairies ou certains ajouts aisément et c'est un plus non négligeable.

Gama Platform est donc un environnement de développement de modélisation et de simulation reposant sur un système multi-agents. Ce système repose sur une base
Java de java, ce qui est un très bon avantage car java est un langage orienté objet très utilisé. Cela permet de modifier des choses dans le code de Gama plus facile que si l'environnement était sur la base d'un langage peu commun ou peu utilisé. C'est un environnement qui propose beaucoup de fonctions déjà intégrées et beaucoup de fonctionnalités très intéressantes qui permettent de pouvoir faire des simulations plus

ou moins réalistes assez aisément. Cela est notamment dû au fait que la documentation est assez bien renseignée et assez précise. De plus, lorsqu'on installe l'environnement, un bon nombre de modèles s'installent en même temps. Ce sont soit des tutos, soit des démonstrations de certaines fonctionnalités proposées par Gama, et ces modèles-là peuvent être très utiles que ce soit pour comprendre un principe, apprendre un principe ou même avoir de nouvelles idées pour optimiser ou améliorer son code grâce à des fonctionnalités proposées par l'environnement. Toutes ces raisons font que Gama est un très bon choix d'environnement de développement.

familier

Programmer en système multi-agents n'est pas comme programmer en orienté objet, en effet, il y a certaines spécificités. Tout d'abord les fichiers de type Gama sont des modèles, comme on pourrait retrouver en programmation par contraintes. Les fichiers de type Gama sont des fichiers d'extension .gaml.
Voici un schéma de l'architecture générale d'un modèle Gama suivi de son explication détaillée.

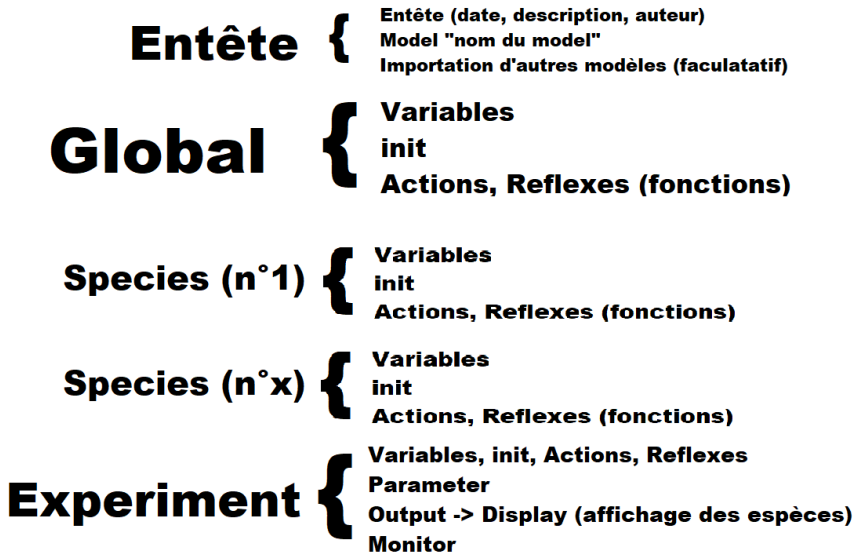


Figure n°1 : Schéma de l'architecture d'un modèle Gama

Ces modèles Gama sont donc divisés en plusieurs parties, tout d'abord nous avons la partie où nous pouvons importer d'autres modèles, ce qui peut permettre de compartimenter le code ou bien d'utiliser le code d'un autre modèle. Nous retrouvons ensuite la partie globale, dans laquelle on déclare les variables et fonctions (actions ou réflexes) propres au modèle. Après cette partie, on peut déclarer toutes les espèces et sous-espèces nécessaires à notre modèle avec leurs variables propres et leurs fonctions propres. Il est à noter que la partie globale et chacune des espèces et sous-espèces peuvent ou non posséder une sous-partie nommée init qui permet d'initialiser les valeurs de certaines variables. Il existe également un système de skills

astérique pour les mots du glossaire

que l’on peut attacher aux espèces. Les skills sont littéralement des "capacités", ce sont des compétences que l’on peut ajouter aux agents car elles ont été programmées et intégrées dans Gama par les développeurs. Un des avantages de ce système de skills c’est qu’ils sont programmés en java et qu’il est tout-à-fait possible d’en créer soi-même, ce qui peut être très pratique si on a besoin d’ajouter une capacité qui peut être implémentée dans plusieurs projets par exemple. La dernière partie d’un modèle contient une ou plusieurs experiment, c’est la partie qui représente tout ce qui est visible de la simulation, elle peut contenir des variables, fonctions et une partie init comme les autres parties. Cependant, elle a la particularité de pouvoir contenir une partie display dans laquelle il faut indiquer quels éléments on affiche à l’écran. Finalement on retrouve dans cette partie des monitors qui sont des affichages de la valeur d’une variable et des parameters qui permettent de changer la valeur d’une variable.

éviter les anglicisme si l'équivalent français existe.

Voici un visuel du projet CESAR version 1 :

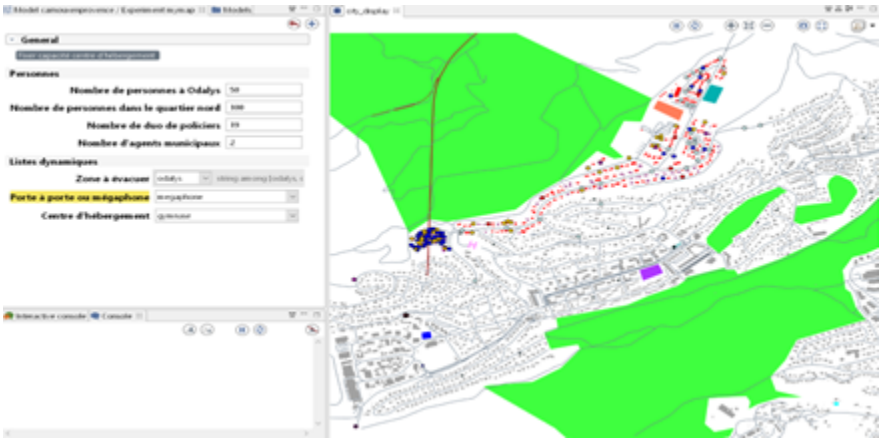


Figure n°2 : Capture d’écran de la simulation CESAR V1

Annexes :

Annexe n°1 : Cahier des charges

Cahier des charges - Lorenzo

Mise à jour des agents existants

-**Ensemble des agents humains** : fonctionnement des déplacements (corriger ce qui existe, ajouter les feux, ajouter la notion de « véhicule » (gestion des collisions et embouteillages), utiliser le correctif intégré à GAMA (cf Philippe)) priorité 1

-**Habitants et touristes** : ajout d'un comportement de « routine » + vérification des comportements (évacuation ou confinement) en fonction des consignes et en l'absence de consigne. priorité 2

Mise à jour des consignes existantes

-Envoyer une consigne : clarification des possibilités (porte à porte, haut-parleur, internet, radio) priorité faible

-Mise à jour des effets de la consigne évacuation priorité faible

-Mise à jour des effets de la consigne confinement priorité faible

-Mise à jour des consignes hébergement (ajout des nouveaux lieux) priorité faible

Ajouts de nouveaux agents

Ajout des habitants des quartiers centre, est et ouest. Priorité moyenne

Ajout de tous les bâtiments d'hébergement de Carnoux et des villes extérieures priorité moyenne

Ajout d'un agent « feu » dont le comportement est simpliste (propagation d'un feu linéaire en fonction de la vitesse du vent) à détails à discuter. Priorité moyenne

Ajout d'un agent « pompier » et d'agents « matériels de pompier ? » Priorité moyenne

Ajout d'un agent « vent » ? Priorité moyenne

Ajout d'un agent « fumée » dont le comportement dépend de l'agent feu Priorité moyenne

Ajout d'un agent « gendarme » Priorité faible

Ajout d'un agent « service technique » Priorité faible

Ajout des agents « outils » Priorité faible

Ajout d'un agent « service logistique » Priorité faible

Ajout d'un agent « bus » et consignes associées Priorité faible

Ajout d'un agent « services médicaux d'urgence » Priorité faible

Ajout des agents du système ferroviaire Priorité très faible

Ajout d'un agent personnel éduc nationale Priorité très faible

Ajout de nouvelles « consignes »

Ajout de plusieurs consignes permettant d'influer sur les segments de routes (vitesse, type de véhicules autorisés et sens de circulation) à réfléchir sur l'ergonomie priorité 3

Ajout de « consignes « permettant de jouer sur le positionnement des pompiers et leurs effets sur le feu Priorité moyenne

Ajout de consigne permettant de livrer du matériel Priorité faible

Ajout de consignes « PPMS incendie activé/désactivé » pour chaque établissement Priorité très faible

Autres

Définition du code couleur affiché à l'écran Priorité forte (à faire au fur et à mesure)

Redéfinition des données paramétrables en début de simulation Priorité forte (à faire au fur et à mesure)