

# Projet d'intelligence artificielle : Simulation de Société d'Agents Coopératifs

Jason CROMBEZ  
Marine LAVAUX  
Rémi PALANDRI

7 janvier 2013

# 1 Sujet

Ce projet consiste à définir une société d'agents où chaque agent aura un rôle. Pour les agents humains ce rôle pourra être :

- cultivateur
- éleveur
- chasseur
- porteurs d'eau
- constructeur
- cuisinier
- chef de clan (un seul agent)

Il y aura également des agents animaux qui pourront être soit sauvages soit domestiquables.

Ce clan et ces animaux vivront dans un univers clos, représentable par une île (grille 2D rectangle), constitué d'une plaine dont une partie sera occupée par le clan (village) et d'une jungle.

Cet univers sera soumis à des intempéries (dont on n'est pas obligé de représenter la nature exacte : foudre, éboulements, inondations, ...) et à des antagonismes (chasses d'animaux qui veulent fuir les humains et survivre à leurs attaques, attaques par les animaux des humains qui veulent fuir et survivre eux aussi, ...) dont l'effet ultime sera toujours la diminution (la mort), dans une certaine proportion (simulée par des tirages aléatoires), du nombre d'agents (humains ou animaux). Les agents auront néanmoins la faculté de se reproduire si certaines conditions sont satisfaites (conditions de ressources alimentaires, ...).

Les agents animaux chercheront essentiellement à se nourrir et à se reproduire. Leur capacité à se nourrir pourra être entravée par le manque d'étendue dont ils peuvent profiter (surface de l'île moins celle du village et des cultures). Leur capacité à se reproduire pourra dépendre de leur capacité à se nourrir et de leur nombre (plus ils sont nombreux plus ils peuvent se rencontrer).

Les objets abris peuvent servir à faire diminuer les probabilités de mort des agents humains, les huttes de stockage servent de réserves et permettent de survivre même si le temps ne permet ni les récoltes, ni la chasse. Ces objets peuvent être détériorés et devront être reconstruits ou réparés.

Toutes les activités prennent du temps et de l'énergie vitale à tous les agents (humains ou animaux). Ils doivent tous se nourrir et se reposer. Ces caractéristiques peuvent aussi être simulées par différentes jauges.

L'agent chef de clan est un peu particulier dans la mesure où il pourrait décider de certaines affectations comme décider combien d'agents de tel type il faut consacrer à la tâche qui leur incombe, décider quels tâches accomplir, combien de temps y consacrer, etc

(voir quels sont les paramètres utiles et sur lesquels le chef de clan pourra jouer).

L'unité de surface cultivable rapporte une certaine quantité de nourriture végétale pour autant qu'elle soit cultivée pendant un certain temps par un certain nombre d'agents cultivateurs. Il en est de même concernant l'élevage des animaux domestiques.

## 2 La technologie utilisée et les sources

### 2.1 Langage utilisé

Le langage utilisé est le Python dans sa version *Python 3.3*. L'interface graphique a été réalisée avec la bibliothèque Qt version *Qt 4.9*.

### 2.2 Comment lancer le projet ?

Une solution consiste à les télécharger et les installer, il suffit de trouver la version correspondante sur leur site officiel et de lancer *main.py* comme tout programme Python.

L'autre solution consiste à utiliser les exécutables *main.exe* clef en main. Il en existe deux, l'un pour du 32 bits et l'autre pour du 64 bits. Ces exécutables ont été testés et fonctionnent sans nécessiter aucun réglage préalable. Donc, sauf si vous voulez re-exécuter le code, il vous suffit de double-cliquer sur le *main.exe* qui convient à votre machine, l'application prend un peu de temps pour se lancer.

### 2.3 Organisation des sources

Les sources sont dans le dossier principal avec comme extension ".py", et le dossier *resources* contient les images utilisées pour l'interface homme-machine.

## 3 Analyse du sujet et modélisation

### 3.1 L'univers

L'univers est clos, il s'agit d'une île. Celle-ci est générée automatiquement à l'aide de Perlin Noise, c'est également le cas en ce qui concerne les différents types de terrains et les ressources naturelles (forêts, lacs, baies, ...) les coefficients peuvent être ajustés afin de créer des cartes ayant leur spécificités propres.

Le perlin noise est un algorithme de génération de bruit 2D qui nous permet d'obtenir des formes continues. Plusieurs couches de perlin noise ont été utilisées pour créer de façon réaliste la carte. La première génère la terre, la plage et l'eau, la 2ème génère les forêts et la 3ème génère les moutons, loups et baies.

La carte est divisée en cellules unitaires, celles-ci contiennent les informations relatives au terrain, aux agents et bâtiments présents sur la cellule. Ces informations sont réparties en deux catégories, celles qui sont mutuellement exclusives (une cellule ne peut être que d'un seul type de terrain) et celles s'ajoutent les unes aux autres (une cellule peut à la fois contenir un bâtiment, plusieurs agents animaux et plusieurs agents humains).

Une cellule étant unitaire, un agent quand il se trouve sur une cellule l'occupe en totalité et n'occupe qu'elle, en revanche une cellule peut-être par plusieurs agents simultanément. Les bâtiments occupent une seule cellule, les cellules sont occupées par un seul bâtiment au maximum. Les agents peuvent traverser une cellule occupée par un bâtiment.

### 3.2 L'interface homme machine

L'interface homme machine comporte plusieurs boutons, expliqué ici (de gauche à droite).

- Quitter : permet de quitter l'application. Si vous avez regénéré la map il se peut qu'il ne marche pas. Dans ce cas alt+F4 ou toute autre méthode marche très bien.
- Configuration : permet de modifier les paramètres pour la prochaine configuration. Mais si on ferme l'application la configuration par défaut est remise.
- Play : lance l'application.
- Pause : met l'application en pause.
- Generate Map : sur windows le bouton est invisible mais il existe et malheureusement il ne régénère pas les acteurs, autrement dit il ne fonctionne pas correctement.
- Zoom + : permet de zoomer.
- Zoom - : permet de dézoomer.

La carte générée a un code couleur pour indiquer les objets et les acteurs.

- Mer : pixel bleu foncé
- Lac : pixel bleu clair
- Plage : pixel beige
- Bois : pixel vert foncé

- Plaine : pixel vert clair
- Baies : pixel rouge
- Montage : pixel gris
- Herbe mangée : pixel noir
- Loup : dessin d'un loup
- Mouton : dessin d'un mouton
- Humain : dessin d'un humain
- Forum : un "F" sur fond blanc

### 3.3 Les bâtiments

Il existe plusieurs types différents de bâtiments.

- le Forum : il sera présent initialement sur la carte et les agents humains auront son emplacement en mémoire, car c'est là où se situera le chef et où viendront les agents pour recevoir des directives.
- Les stockages de ressources : ils sont spécialisés dans un type de ressource particulière et ont une capacité maximale.
- Les abris : ont une capacité maximale, la mémoire des agents présents dans le bâtiment n'est pas assurée par le bâtiment mais par l'agent.
- Les champs : ils sont comme des stockages de nourriture dont le contenu augmenterait périodiquement si ils sont cultivés.
- Les chaudrons : ce sont les endroits où les agents humains dont le rôle sera de cuisiner cuisineront et donc où les agents humains ayant faim viendront se restaurer.

### 3.4 Les agents

Qu'ils s'agissent des animaux ou des humains, la modélisation de ceux-ci à une racine commune. Il s'agit de la classe *Etre* dans le fichier *etre.py*. Cette représentation dont hérite tous les types d'agents regroupe ce qui peut être factorisé. Il s'agit pour une part de tout ce qui concerne la gestion des entités en lien avec l'interface et d'autre part d'attributs non spécifiques à un type d'agent particulier (le genre par exemple, qui sera utilisé dans le cadre de la reproduction ou bien l'état de l'agent, vivant ou mort).

Tous les types d'agents disposeront d'une fonction *run* qui sera jouée à chaque tour de la simulation, certains comportements tels que la mort si la jauge de nourriture tombe à zéro seront également universels.

La modélisation du vieillissement est elle aussi commune, l'âge en lui-même n'est pas pris en compte mais la mort des agents par vieillesse existe. La modélisation mathématique retenue est une simplification du modèle de Weibull (courbe en baignoire). En effet au début le taux de mortalité est décroissant et correspond à une période de mortalité infantile, ensuite le taux de mortalité est constant (et faible) puis il ré-augmente.

Dans notre modèle la courbe a été simplifiée en la modélisant par des droites dont les coefficients ont été choisis arbitrairement, cette modélisation se trouve dans le fichier *LifeExpectancy.py*. A chaque appel de la fonction *run* pour savoir si l'agent meurt de mort

naturelle.

Par ailleurs dans le cas où l'agent devra chercher sur la carte quelque chose dont il ne connaît pas a priori la position, la recherche se fera mais dans une zone restreinte centrée sur l'agent, et ceci afin de modéliser ce qu'il peut voir d'où il est.

### 3.4.1 Les animaux

Il existe deux types d'agents animaux, les domesticables et les sauvages. Sur l'interface graphique ceux-ci sont représentés respectivement par des moutons et des loups.

Les éléments factorisables à ces deux types d'agents sont regroupés au sein de la classe *Animal* se trouvant dans le fichier *Gaia.py* et héritant de la classe *Etre*. Les classes *Sheep* et *Wolf*, également présentes dans *Gaia.py* héritent de cette classe *Animal*. Ce type d'héritage multiple étant possible en *Python*. Ce que l'on factorise au niveau de la classe *Animal* consiste essentiellement en des comportements (méthodes). Il s'agit des fonctions concernant la reproduction.

Comme précisé dans le sujet, la reproduction dépend de la possibilité de deux animaux de sexe opposé à se rencontrer ainsi que du fait que ceux-ci sont suffisamment bien alimentés. Les fonctions nécessaires sont donc de rendre possible la vérification de ces conditions (en terme de genre et d'état de la jauge de nourriture), ainsi que la recherche d'un partenaire pour se reproduire si ces conditions sont vérifiées, le fait de le rejoindre ainsi que l'enfantement.

Dans le cadre de la modélisation de ces comportements nous avons fait certaines hypothèses, celles-ci pouvaient être simplificatrices ou bien avoir un caractère arbitraire. L'hypothèse simplificatrice que nous avons faite est de considérer que le temps de gestation est nul, de plus l'individu qui voit le jour est instantanément considéré comme étant un adulte. Les choix arbitraires que nous avons faits se situent au niveau de tous les paramètres qu'il nous a fallu fixer, il s'agit par exemple du champ de vision des animaux, le rayon de ce champ de vision est une constante arbitraire et il en va de même pour le niveau de la jauge de nourriture qui définit si l'animal a faim ou bien est disposé à se reproduire.

Les animaux ont tendance à rester en groupe, et ça pour plusieurs raisons. La première est due à leurs déplacements permanents. En effet, même si ils ne cherchent pas à se reproduire, ils se déplacent de temps en temps. Ces déplacements se faisant aléatoirement, avec équiprobabilité pour chacune des directions, il est donc normal qu'après une infinité de déplacements ils ne bougent pas. Par ailleurs, quand pour une raison quelconque (à l'initialisation, par exemple) ils ne sont pas en groupe, la recherche de conjoint va faire que des petits groupes d'animaux finiront par se former.

#### – Les moutons

Les moutons se nourrissent d'herbe, celle-ci se trouve sur les terrains de type plaine, l'herbe de la cellule est alors consommée et met un certain temps à repousser. Ils ne peu-

vent donc pas manger deux fois de suite au même endroit.

La fonction *run* du mouton dépend essentiellement de l'état de sa jauge de nourriture si il est assez alimenté il cherchera à se reproduire si ce n'est pas le cas soit (cette condition est déterminée par le hasard mais pondérée par la jauge de nourriture) il se mettra à errer ou à se nourrir (la probabilité qu'il se nourrisse sera d'autant plus importante que la jauge de nourriture sera basse mais dépend en partie d'un nombre arbitraire).

Les moutons ne communiquent entre eux que pour savoir s'ils acceptent ou non de se reproduire.

Le décès du mouton peut survenir par le fait que leur jauge de nourriture tombe à zéro mais ceci est loin d'être leur principale cause de décès.

#### – Les loups

Les loups se nourrissent de moutons et d'humains, qu'ils doivent donc chasser. La principale différence dans la fonction *run*, par rapport, à celle du mouton, est donc que de même qu'il y a recherche dans le cas de la reproduction, il y a également recherche lorsqu'il faut se nourrir.

La fonction de recherche de nourriture du loup contrairement à celle de reproduction des animaux prend en compte le périmètre de vision du loup. Le loup quand il s'agit de manger n'a pas une vision globale de tous les animaux de la carte mais seulement ceux dans un petit rayon. Il est aussi important de noter que le loup choisira l'animal le plus proche, puis le plus gros parmi ceux là. Tous les loups ne sont pas égaux, ils ont tous une force différente, indépendante du sexe. La force leur permet de déterminer s'ils vont gagner leurs combats contre leurs proies. Quand le loup gagne le combat il mange sa proie et se nourrit en fonction de la jauge de nourriture de la victime. Mais s'il perd, le mouton fuira ou l'humain le tuera. Le vieillissement des loups est modélisé par la perte de force du loup. A chaque combat le loup perd de la force. Les loups sont des agents non communiquant, mises à part lors de la reproduction tout comme les moutons.

Les principales raisons de décès des loups sont leur positionnement initial (trop loin des troupeaux de moutons) ainsi que leur mauvaise gestion des moutons, en effet ils déciment bien trop rapidement les troupeaux de moutons.

### 3.4.2 Les humains

*La spécification suivante est une proposition d'interprétation du sujet*

Les agents humains disposent de deux jauges qui doivent être au dessus de zéro pour que l'agent survive. Il s'agit de la jauge de nourriture, qui baisse en continu de façon indépendante des activités et la jauge de fatigue qui, elle baissera en fonction de l'activité pratiquée par l'agent. Pour remplir la jauge de nourriture l'agent pourra consommer de la nourriture issue de la chasse, de la cueillette, de l'agriculture ou de l'élevage. S'il consomme



directement ces aliments ceux ci seront moins nourrissants que s'ils sont préalablement cuisinés par un agent cuisinier. Afin de remplir sa jauge de fatigue l'agent peut cesser l'activité qu'il pratique et se reposer, c'est pendant cette période de repos qu'il aura l'occasion de se reproduire.

Mis à part le chef dont le système de mémoire est plus développé, celui des autres agents reste relativement limité. Deux agents se trouvant sur la même cellule partagent les informations dont ils disposent avec d'autant plus de probabilités que l'information sera en rapport avec l'activité courante de l'agent.

- Le chef

Le chef est unique, il est chargé de donner des ordres aux autres agents en tenant compte des connaissances dont il dispose concernant l'état des ressources dont dispose le clan. Les tâches à effectuer sont a priori en temps limité et le chef considérera donc comme mort les agents ne revenant pas demander du travail au bout d'un certain temps.

Le chef disposant d'une mémoire plus complète que les autres agents, il est un recours pour ces derniers, en effet si ils sont pris en défaut car ils n'arrivent pas à trouver quelque chose par exemple ils peuvent demander au chef de leur indiquer si celui ci dispose d'informations supplémentaires.

- Les constructeurs

L'agent constructeur se voit assigner comme tâche par le chef de construire un bâtiment précis, il lui revient donc de trouver un emplacement adéquat, c'est à dire une plaine qui ne soit pas déjà occupée par un autre bâtiment, de construire le bâtiment en question (le nombre de tours nécessaire dépendant du bâtiment) et de retourner voir le chef pour lui indiquer l'emplacement du nouveau bâtiment et recevoir une autre tâche à accomplir.

- Les cultivateurs

Le cultivateur, quand il se voit attribuer cette tâche, se fait indiquer par le chef l'emplacement d'un champ (considéré comme un bâtiment) préalablement construit et non cultivé par un autre agent. Le champ dispose d'une capacité maximale et sa quantité de nourriture augmente périodiquement pour peu qu'il soit cultivé. C'est le rôle de l'agriculteur de cultiver son champ et d'aller quand celui ci est plein apporter la nourriture dans un stockage adapté.

- Les cuisiniers

Le cuisinier, quand il se voit attribuer cette tâche, se fait indiquer par le chef l'emplacement d'un chaudron préalablement construit et non occupé par un autre agent. Le chaudron dispose d'une capacité maximale et les agents souhaitant se nourrir pourront venir s'y restaurer de façon plus efficace que s'ils cosommaient de la nourriture directement. C'est le rôle du cuisinier d'aller chercher de la nourriture dans un stockage quand

son chaudron et vide. On considère que le temps de préparation sera pas pris en compte et donc que la nourriture sera donc immédiatement accessible une fois dans le chaudron et que la nature de la nourriture (viande, baie, issue de l'agriculture ou de l'élevage) n'aura pas d'incidence sur le caractère nourrissant ou non de ce qui est préparé.

– Les chasseurs

Le chasseur pourra éventuellement se faire attribuer une zone de chasse recommandée par le chef lors de l'attribution de cette tâche, dans le cas contraire il devra soit piocher dans sa mémoire pour savoir si il connaît une zone de chasse potentielle, soit explorer afin de découvrir une zone où il voit un mouton ou un loup. Son comportement sera différent ensuite selon qu'il a à faire à un mouton (animal domesticable) ou à un loup (animal sauvage). Si l'animal est un mouton le chasseur, une fois sur une cellule adjacente, tentera de tuer le mouton (avec une probabilité arbitraire de réussite) en cas de réussite il ramène la viande au village, en cas d'échec le mouton s'enfuit hors d'atteinte. Dans le cas des loups ils ont un comportement différent que les autres agents, alors que les autres humains ne font que se défendre contre les loups, les chasseurs vont plus loin et ne laisse pas le cadavre par terre, mais prennent la viande pour nourrir le village.

Les chasseurs permettent donc au clan l'accès à la nourriture. Ils trouvent leur place entre les cueilleurs et les éleveurs. La denrée qu'ils fournissent est très rentable mais aussi dure et risquée à se procurer. Ils sont un boost énorme pour l'économie du village. Si le chef en abuse il empêche l'élevage alors que l'élevage permet au village d'avoir une économie stable, permettant de se concentrer plus sur les autres points de la ville.

– Les éleveurs

L'éleveur, quand il se voit confier cette tâche, se fait indiquer par le chef l'emplacement d'un groupe de moutons. Son rôle sera d'aller en chercher, les nourrir, en donnant la priorité à ceux ayant la jauge de nourriture la plus basse, afin que ceux ci se reproduisent. Il devra faire un rapport régulier au chef en lui disant combien de moutons sont sous sa garde (en particulier s'il y a eu naissance) et pour savoir s'il doit continuer à les nourrir, les laisser simplement paître (et donc faire une autre activité qu'éleveur) ou bien abattre certains animaux afin d'augmenter les stocks de nourriture.

– Les cueilleurs

Les cueilleurs sont les premiers fournisseurs de nourriture du clan. Ils récoltent lentement de la nourriture sur les baies. Les baies disparaissent après leur récolte pour devenir de la plaine. Elles repoussent après un certain laps temps. Une tribu n'ayant qu'accès sur les baies ne peut pas évoluer. Le seul avantage des baies c'est de ne présenter aucun risque à récolter et de ne nécessiter aucune infrastructure particulière.

– Les porteurs d'eau

Les porteurs d'eau vont chercher de l'eau douce pour la tribu. Leur rôle est indispens-

able, l'eau est une denrée clé pour la ville. Il est impossible d'améliorer leur rendement. Le chef est donc obligé d'en demander constamment et de façon linéaire. Plus le forum est loin d'un point d'eau douce plus la ville mettra du temps à grandir.

Cette ressource est considéré comme étant inépuisable et n'entraîne donc de diminution d'un stock, les lacs ne se vident pas.

- Les bûcherons

Les bûcherons sont le pilier de l'avancement de la tribu, ils permettent l'accès à la construction de bâtiments, car ceux ci nécessitent du bois. Le bois comme les baies disparaissent après avoir été coupé et repousse avec le temps. On peut vivre sans bois, mais on ne peut pas bien croître sans. Le chef doit penser à eux quand la nourriture et l'eau sont en croissance.

- Les explorateurs

Le chef utilise les explorateurs au début, afin de mieux connaître l'environnement autour du village. Il va leurs indiquer une direction, informatiquement ça revient à demander à l'explorateur de pondérer ses déplacement pour qu'au final il prenne une certaine direction qui va permettre de connaître un endroit.

L'explorateur ne va pas enregistrer toutes les informateurs mais uniquement les informations importantes comme la position de baies, du bois, de l'eau douce. Pour indiquer au chef la position des moutons, l'explorateur va définir ce qu'est un groupe de moutons et enregistrer uniquement le centre de leur groupe. De cette façon l'éleveur et le chasseur n'auront qu'à y aller puis chercher dans leur champ de vision.

Cette gestion de l'explorateur permet de ne pas encombrer inutilement la mémoire. L'explorateur doit bien faire attention à ses jauges pour éviter de mourrir. Il doit donc prendre en compte le temps de retour.

### *La suite présente l'état actuel de nos agents humains*

Nous avons implémenté cinq des rôles que nous aurions voulu avoir. Les rôles des humains sont définis à leur création et ne seront pas changé lors de l'exécution (mais ils sont changeables).

- Les ramasseurs

Les rôles de type ramassage de ressource ont été implémentés : il s'agit des cueilleurs, des porteurs d'eau et des bûcherons. Ces trois rôles ont pour but de ramener des matériaux de la source au forum. Cela permet d'avoir une interaction entre l'environnement et les humains. Les humains interagissent donc avec les baies, le bois et les loups. De plus les moutons interagissent avec les loups et l'herbe. Au final tous les éléments créés sont reliés entre eux.

– L’explorateur

Il est indispensable pour tous les autres. C’est lui qui va chercher les informations importantes de la carte. Il se déplace selon une direction donnée qui change quand il rentre au forum et ramène au camp les informations. Il enregistre la position des baies, du bois et des sources d’eau douce. Il a un compteur de fatigue, dès qu’il est fatigué il rentre au forum pour se reposer et rendre compte de ses trouvailles. Quand un obstacle le bloque, il change aléatoirement de direction.

– Le chef

Le chef a pour rôle d’être la mémoire globale du village, il gère aussi la reproduction. Il reste sans bouger au forum et est immortel. Il recueille les informations de tous les autres agents humains. En effet les explorateurs prennent tout type d’informations alors que les trois autres enregistrent uniquement les informations sur la position de leur type de ressource. Par exemple, un bûcheron dira au chef la position de tous les bois qu’il a vu. Quand ils rentrent au forum ils oublient tout et le chef gère la ressource afin d’éviter les doublons.

Il y a un autre type de mémoire. Elle permet une gestion en aval et évite d’avoir des informations erronées. Les récolteurs vont détruire les ressources et il serait gênant qu’un explorateur dise au chef qu’en fait celle-ci existe encore. C’est pour cela que le cueilleur et le bûcheron vont dire au chef quelles ressources ils ont détruites. Le chef va garder dans la mémoire la destruction de cette matière première. Par contre il faut le dire au chef en même temps qu’il assigne une destination aux récolteurs, sinon deux récolteurs risquent d’aller au même endroit. Cette solution crée un problème, si le récolteur meurt l’information du chef sera erronée, mais cela reste une solution acceptable. Le chef garde cette information durant le double de la durée de fatigue de l’explorateur.

Les récolteurs demandent au chef la première destination qu’il leur propose, l’implémentation pour avoir la plus proche à été faite mais elle est trop lourde.

L’autre rôle du chef est de gérer la population. Quand la nourriture est au-dessus d’un certain seuil il donne naissance à un nouvel humain. Le nouvel humain a plus de chance d’être un cueilleur puis un explorateur et enfin autant de chance pour les deux autres rôles possibles.

Le seul bâtiment utilisé est le forum, il a une capacité maximum pour l’eau, le bois et la nourriture. Les autres types de bâtiments existent mais pas les agents constructeurs.

L’état initial est le suivant : un forum et un humain de chaque rôle.

Au début seul l’explorateur bouge, les autres attendent des informations de déplacements. Puis au retour de l’explorateur tous les humains se mettent à bouger.

## 4 Bilan

Les résultats concernant les agents animaux sont à la hauteur de l'analyse que nous avons faites, car ils sont implémentés d'après l'analyse et la spécification que nous avons créée à partir du sujet. En revanche, seule une partie de ce que nous avons défini concernant les agents humains a été implémentée. Ceci étant pour une part dû à une évaluation erronée de la difficulté à implémenter les concepts et la modélisation que nous avons tirée du sujet, et d'autre part car la méthode de codage employée n'était pas adaptée. En effet, croyant qu'il allait être possible de coder tout ce que nous avons défini, nous avons essayé de coder tout ce qui concernait les humains d'un coup. Rétrospectivement on peut se rendre compte que cela nous a fait perdre beaucoup de temps et d'efficacité. C'est en changeant de méthode et en utilisant une méthode de codage plus par itérations successives que tout en bloc que nous avons réussi à coder quelques comportements pour nos agents humains.