Intelligence Artificielle pour le Jeu Vidéo
Support de cours n°5

Logique floue

Contact : Agnes Delaborde – agnes.delaborde@gmail.com

Dates
Dimanche 10 avril : envoi du code et du rapport (par mail) **Mercredi 13 avril** : envoi de la présentation (par mail) **Vendredi 15 avril** : soutenance

Objectifs

Évolution de dimensions en logique floue.

Exercice 1 – Patience et longueur de temps...

Nous allons intégrer la dimension « patience » dans la simulation. La patience de Gus se manifeste de la sorte : plus il reste longtemps sans rien faire, moins il est patient. Dès qu'il effectue une action, son niveau de patience revient à son maximum. Pas de logique floue dans cet exercice.

a. Récupérez le fichier patience.lua dans le dossier partagé.

b. Suivez les instructions mises en commentaire dans le fichier : les premières lignes doivent être mises dans constants.lua, et le reste du script peut être rangé dans gus.lua.

   Soyez vigilants que les noms de variables ne rentrent pas en conflit avec de nouvelles variables de votre script.

c. Créez une nouvelle jauge affichant la valeur de Gus.patience.

   Pour rappel :

   1. Créez une instance de Gauge dans love.load().
   2. Créez les objets Image associés (le caption_img est disponible dans /assets/img/), dans love.load().
   3. Appelez la méthode gauge_patience:draw() depuis love.draw().
   4. Éditez Gauge:draw() afin de prendre en compte le label GUS_PATIENCE_LABEL.

d. Nous allons mettre à jour le timer de la dimension patience à chaque tick du jeu : appelez Gus.setPatienceTimer() à la fin de love.update() (qui est dans main.lua).

e. Le timer est réinitialisé, et la valeur de Gus.patience remise à son maximum, grâce à la fonction Gus.resetPatience(). Déterminez (et implémentez) à quel(s) moment(s) vous devez appeler cette fonction.

f. Combien de temps faut-il avant que Gus ait perdu toute sa patience ? Trouvez la variable correspondante, ainsi que sa valeur par défaut.

   C'est cette valeur que nous allons mettre à jour via un système de règles en logique floue.

Exercice 2 – Ça reste quand même très flou pour moi…

Vous allez étudier le système en logique floue permettant de mettre à jour la variable Gus.patienceTimer.waitingTime.

a. Sur le dossier partagé, récupérez le script fuzzylogic.lua et mettez-le à la racine de votre dossier de jeu.

b. Dans main.lua, chargez le fichier fuzzylogic.lua juste après le fichier gus.lua.

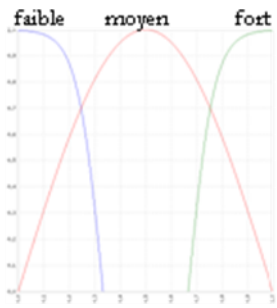
Le script associé à cet exercice nécessite que vous ayez toujours les dimensions **Gus.vigor** et **Gus.sociability** dans votre simulation.

- Si vous les avez renommées/réorganisées : éditez les noms de ces variables dans le script fuzzylogic.lua (vérifiez également que GUS_SOCIABILITY_MAX, GUS_SOCIABILITY_LABEL, etc. sont toujours d'actualité) ;
- Si vous les avez supprimées : adaptez les règles floues avec d'autres dimensions ayant du sens (conceptuellement) pour le calcul de waitingTime. Préférez des variables continues plutôt que discrètes.

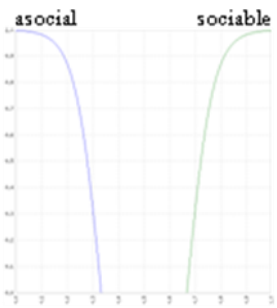
N'hésitez pas à m'appeler pour réaliser les modifications !

c. Dans ce script, vous voyez qu'une instance de FuzzyLogic a été créée, « FL_patienceUpdate ». Voici la représentation graphique des fonctions d'appartenance pour les variables linguistiques de Vigueur et Sociabilité (input), et Temps d'attente (output).

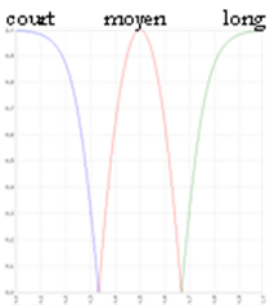
Représentation graphique des fonctions d'appartenance
(abscisses : valeur normalisée [0-1] ; ordonnées : degré d'appartenance [0-1])



Vigueur



Sociabilité



Temps d'attente

Ces fonctions sont déclarées dans FL_patienceUpdate.membershipFunctions.

d. Dans FL_patienceUpdate.decisionRules, vous trouvez 3 règles. Elles sont résumées dans ce tableau :

Valeur de vigueur	Opérateur	Valeur de sociabilité	Impact sur Temps d'attente
Si Gus est faible…	…ou …	… s'il est d'humeur asociale,	alors il attendra très peu de temps.
Si Gus est en moyenne forme,			alors il attendra moyennement longtemps.
Si Gus est en pleine forme…	…et…	… s'il est sociable,	alors il attendra longtemps.

e. Un peu de simulation à la main. Pour chaque situation ci-après, essayez de déterminer dans quel sous-ensemble de Temps d'attente (court, moyen, ou long) la valeur de sortie se trouvera :

1. Si Gus.vigor = 13 et Gus.sociability = 13.
2. Si Gus.vigor = 100 et Gus.sociability = 100.
3. Si Gus.vigor = 50 et Gus.sociability = 100.

f. Réactivez les lignes commentées à la fin du script fuzzylogic.lua, et testez avec les valeurs données à la question précédente. (N'oubliez pas de commenter à nouveau ces lignes ensuite)

g. A chaque fois que Gus se déplace (dans Gus.movement.move()), réalisez cet appel :

Gus.patienceTimer.waitingTime = FL_patienceUpdate:makeDecision(_, norm)

comme nous l'avons vu à la question précédente.

h. Lancez la simulation, et observez les variations de comportement de la patience de Gus occasionnées par ses activités.

Projet – Propositions relatives à ce cours

Proposition 1 : **[en logique floue]** La couleur de Gus variera en fonction de son niveau de vigueur et de patience.

Un exemple de règle : s'il est faible et très patient alors il sera gris foncé.
Pour changer la couleur de Gus, utilisez love.graphics.setColor(x,x,x) avant de dessiner Gus dans love.draw().

Proposition 2 : **[en logique floue]** Les femmes illuminent les journées de Gus. Créez un système flou intégrant ces règles :

Si Gus a rencontré beaucoup de fois le PNJ femme, et s'il a vécu beaucoup de situations satisfaisantes[1] (voir TD2), alors Gus sera heureux.
Si Gus a rencontré peu de fois le PNJ femme, et s'il a vécu beaucoup de situations satisfaisantes, alors Gus sera moyennement heureux.
Si Gus a rencontré très peu de fois le PNJ femme, alors Gus sera malheureux.

Note [1] : pas nécessairement seulement avec elle, mais avec tous les PNJ