19/06 :

Premier jour, présentation du projet par les encadrants, plusieurs choix s’offrent à nous. Nous pouvons soit utiliser une interface qui a été faite par un groupe l’an passé sur netlogo et donc nous focaliser uniquement sur les paramètres de modélisation et leurs impacts. Il est également possible de recréer notre interface graphique sur netlogo ou processing. Jugeant que la tâche de créer l’interface graphique et la simulation n’était non seulement pas insurmontable mais qu’elle était surtout intéressante à faire pour apprendre à programmer. C’est pourquoi nous avons pris la décision de tout programmer nous même sur python en utilisant la bibliothèque pygame pour l’interface graphique utilisateur.

Une fois le choix de l’approche, nous avons établi un cahier des charges pour savoir la direction vers laquelle nous allons mais surtout pour nous assurer d’avoir bien compris le sujet ainsi que ses enjeux, et donc ce que nous cherchions à démontrer/comprendre.

Voici donc les éléments qui nous semblaient les plus importants :

-des spots de nourritures dont nous décidons la position, l’attractivité, le nombre, la qualité de la nourriture

-des agents, qui possèdent une énergie, un type (altruiste, tricheur, basique), des gènes, une capacité à se reproduire (seuls ou à plusieurs), une “intelligence” (aller vers les phéromones/la nourriture)

En parallèle nous avons établi l’ensemble des éléments que l’utilisateur (nous) peut modifier :

-part d’altruistes/basiques/tricheurs au début de la simulation

-nombre de spots de nourritures, quantité de nourriture par spots, attractivité de la nourriture

-énergie de base des agents, énergie nécessaire à la reproduction, coût en énergie de la reproduction

-aging process, combien d’énergie les agents perdent en fonction de leur âge à chaque itération

-puissance des phéromones

-probabilité de mutation des gènes

-choix du gène dominant entre altruiste/tricheur (le gène basique est totalement récessif pour qu’il ne puisse pas proliférer)

Nous avons également pensés à certains ajouts qui pourraient être faits par la suite :

-collisions entre agents

-topologie de la nourriture (empoisonné → les altruistes préviennent, les autres non)

Une fois les tâches déterminées, nous avons séparé le travail comme suit :

-Achraf et Chems s’occupent du déplacement des agents

-Romain s’occupe de l’interface graphique et de coordonner l’univers (implémentation entre les différentes classes)

-Joseph s’occupe des spots de nourriture

-Yanis s’occupe de la reproduction des agents (seul ou à plusieurs) et de l’alimentation des agents

Dès lors, la programmation commença.

20/06 :

Ayant bien segmenté le travail, nous pûmes commencer à intégrer les différentes classes dès le deuxième jour. La première simulation de déplacement des agents suivaient une trajectoire rectiligne uniforme et dès que l’agent arrive face à un mur alors il change aléatoirement de direction. Très vite Achraf s’est penché sur la question d’un mouvement plus “naturel” des agents, il a essayé de faire une marche “aléatoire”, ce qui s’est révélé compliqué. Premièrement, toutes les classes se sont retrouvées dans un même fichier, après réflexion, nous avons fait un fichier par classe pour mieux pouvoir modifier et comprendre le programme. Étant donné que nous avions beaucoup appuyé la veille sur l’importance d’un code commenté (en anglais), il n’était pas compliqué de comprendre ce que les autres avaient fait. Après le passage et les conseils de monsieur Dessalles, nous avons mis toutes les constantes du programme dans un fichier json pour que l’on puisse modifier les constantes simplement avant de lancer le programme. Pour éviter les répétitions dans les différentes classes, une classe mère “entity” a été créée.

Nous avons passé l’après-midi à mettre en commun ce que nous avions fait la veille au soir et le matin pour avoir une ébauche de simulation fonctionnelle. Le travail de Chems et Romain le soir ont permis d’avoir cette embauche fonctionnelle.

21/06 :