

Voici le résumé de la première réunion, le 27 janvier.

Tout d'abord, petite précision et chose à faire rapidement :

- trouver un article à lire sur le multi-agents
- le rapport sera fait en Latex
- il faut créer un dépôt GIT du Sif pour mettre le travail et le rapport

Résumé du projet :

Nous allons développer un système multi-agent sur une même machine, pas de réseau.

Nous avons d'abord parler de faire un ordonnanceur de tâche pour exécuter les agents un à un, comme en NetLOGO, puis nous avons plutôt opter pour l'exécution en parallélisme des tortues, car faire un ordonnanceur causerai des difficultés, (comme comment récupérer la main sur les tortues(utilisation de alarm, mais on ne sait pas qui, ect),et nous préférons utiliser les mécanismes UNIX existant. Nous utiliserons des threads pour programmer les processus tortues plutôt que les forks car ils sont plus modernes, et nous permettront l'usage de moyens de communication comme la mémoire partagée (vs les files de msg avec fork), une meilleure performance, et pour la synchronisation il y aura les mutexs (vs les sémaphores avec fork).

Chaque thread devra interpréter son code.

On aura un thread_main qui aura ses instructions, et la main. Il sera le thread pré-existant, et il y aura des thread_turtles pour les tortues. Nous ferons un MVC, et utiliserons Qt pour l'interface, et C++.

Nous allons procéder de manière incrémentale, avec la méthode AGILE. C'est à dire que nous commencerons par avoir un noyau : une tortue qui s'affiche sur l'interface graphique 2D et qui peut executer 3 instructions simples. Pour commencer, le plan aura une taille fixe. Nous améliorerons l'interpréteur au fur et à mesure et nous ferons des intégrations successives des fonctionnalités.

A faire :

- Quels sont ses moyens de communication des agents ?
- Réfléchir si 1 agent = 1 thread ? thread séquentiel ? système de jeton ?
- Les agents peuvent-ils communiquer avec les patches ? Avec qui peut-on communiquer ?