

Zoo distribué

Mini-projet

Programmation réseaux, concurrente et distribuée

Caractéristiques

L'objectif de ce projet est de réaliser un zoo distribué : un ensemble d'objets qui évoluent dans un cadre fermé 2D (le zoo). Ces objets "vivent" sur des machines différentes, mais le rendu est commun. Cela signifie que lorsqu'on lance le zoo sur plusieurs machines, chaque machine fait évoluer ses propres objets, et affiche à l'écran non seulement ses objets, mais aussi les objets des autres machines, chaque machine devant donc envoyer le nouvel état de ses objets à toutes les autres machines pour qu'elles mettent à jour leur affichage. Si le programme est arrêté sur une des machines, les objets correspondant disparaissent de l'écran des autres machines. Si le programme est lancé sur une nouvelle machine, les nouveaux objets qui évoluent dedans apparaissent.

Zoo sur une seule machine

On traite d'abord le cas d'une seule machine, il s'agit de prendre en main le fonctionnement du zoo.

Les animaux du zoo peuvent se déplacer vers une destination avec une vitesse donnée. La vitesse de déplacement dépend de l'animal. L'affichage est mis à jour périodiquement (toutes les 100 ms par exemple). Par conséquent, tout objet n'est pas arrivé à sa destination, on le fait un petit déplacement en direction de la cible. Cette destination est choisie au hasard. Les animaux changent effectivement de destination dès qu'ils ont atteint celle qu'ils visaient.

Le zoo contient des obstacles. Si un animal rencontre un obstacle, il change sa direction, c'est-à-dire, il choisit une nouvelle destination.

Certains animaux peuvent changer d'avis et choisit de nouveau une destination aléatoire. Lorsque deux animaux de sexe opposé se rencontrent, ils font des petits, qui grossissent petit à petit jusqu'à une taille aléatoire autour d'une taille moyenne donnée. Le nombre des petits

dépend de l'animal. En réseau, les machines possédant père et mère créent chacune quelques petits.

Passage en réseau

Il s'agit maintenant d'afficher les animaux des programmes tournant sur d'autres machines. Dans un premier temps, faites une version simple avec un support pour seulement 2 machines à la fois. Sur une machine, on lance le programme qui doit ouvrir une socket TCP en écoute, il joue le rôle du serveur. Sur l'autre machine, on lance un programme qui doit, dans ce cas, se connecter en TCP à la première machine, il joue le rôle du client. Après l'établissement de la connexion, il faut transmettre les informations sur les objets, c'est-à-dire, il faut établir un protocole d'échange d'objets (des animaux).

Pour passer à plusieurs machines, le plus simple est d'utiliser un serveur centralisé, en réutilisant la version basique de votre serveur, qui va seulement s'occuper de l'échange d'objets.

Le principe du protocole est que chaque machine reste maître du mouvement des objets qu'elle gère. Les autres machines se contentent d'afficher ces objets aux nouvelles positions, lorsqu'elles les reçoivent via le réseau. On utilisera des couleurs différentes pour distinguer les animaux venant de machines différentes.

La réalisation

La réalisation se fera par binôme. Cette règle ne souffrira aucune exception. Bien entendu, chacun dans un binôme devra travailler !!!

La réalisation se fera nécessairement en Java. Chaque groupe devra réaliser le client et le serveur.

D'autre part, il est impératif de respecter les spécifications données dans le sujet sachant que les sections en italique sont des extensions. Bien entendu, si vous souhaitez étendre le logiciel cela est permis et sera jugé très favorablement. Mais vos réalisations devront respecter le cahier de charge de base (c'est un minimum impérativement requis et qui sera testé).

Il est également très souhaitable d'utiliser les outils de conception (diagramme de classe, ...) et les processus de développement de logiciel (scrum, ...).

Note : en cas de doute, consultez votre enseignant pour vous faire une idée. Si vous pensez qu'une imprécision majeure ou une erreur grossière a été commise dans le sujet, prenez contact avec l'enseignant.