

Rapport TP n°3, SE :
Simulation du fonctionnement
d'un cinémas,
Les cinémas GrasMon-Paté

Auteurs : DIALLO, Djiby – KHEIRI, Sarah

Professeur : ROUXEL, Benjamin

Master 1, Génie logiciel, Groupe 1.1

Introduction

Ce TP consistait en la réalisation du fonctionnement d'un cinémas afin de mettre en œuvre les notions de synchronisation de processus vus en cours.

Nous devrions pour cela modéliser le fonctionnement d'un cinémas pour expérimenter les problèmes de synchronisation qui en découlent.

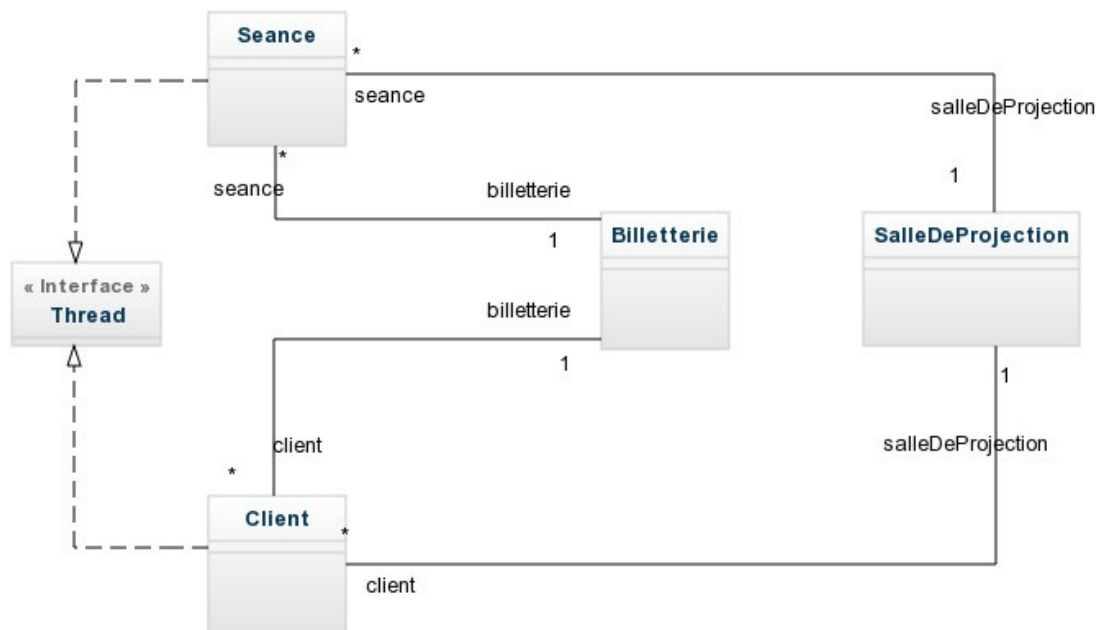
Trois versions ou parties ont été réalisées pour ce TP, comme demandé dans le cahier des charges (Sujet TP), permettant de rajouter à chaque fois de nouvelles fonctionnalités à notre cinémas.

Nous détaillerons dans chacune de ces parties l'organisation de notre cinémas, les problèmes de synchronisation rencontrés ainsi que les solutions trouvées pour les résoudre.

Choix de conception

Les choix de conception dont nous parlerons ici se limiteront seulement sur la partie 1 (Présentation) du TP et s'il y a des modifications dans les autres parties nous ne manquerons pas de les mentionner par la suite.

1 Diagramme de classes



2 Détail du diagramme de classes

Moniteurs de notre système : les classes **SalleDeProjection** et **Billetterie**

Threads de notre système : les classes **Seance** et **Client**

La classe **Cinema**, est notre classe principale.

Partie 1 : Présentation

1 Contexte de simulation

Pour cette partie il fallait simuler un cinéma avec une seule séance pour un film dans une seule salle de projection. Les tickets de la billetterie du cinéma seront vendus aux clients par 1 seul guichet. Si la billetterie n'a plus de places à vendre les clients malchanceux s'en retournent à leurs pénates.

2 Problème de synchronisation rencontrés

Dans cette partie on a pas vraiment rencontré de problème de synchronisation, mais juste une mauvaise gestion des conditions de vente de tickets (Client pouvant acheter un ticket même s'il y en n'a pas).

3 Solutions

Ajout de variables de contrôle et vérification du programme à l'aide de plusieurs tests.

4 Traces d'exécutions

Le client 1 a acheté un billet pour la séance.

Nb de tickets restant: 1

Le client 2 a acheté un billet pour la séance.

Nb de tickets restant: 0

Début de la séance Séance 1

Entrée client 2

Entrée client 1

Fin de la séance Séance 1

Sortie client 2

Sortie client 1

Partie 2 : Développement

1 Un cinéma à plusieurs séances

a) Contexte de simulation

Maintenant les séances se succèdent les unes à la suite des autres. Le cinéma restera ouvert tant qu'il y a des séances à programmer, les employés virtuels peuvent faire des pauses et une séance ne peut débuter que si la précédente est terminée et si tous les spectateurs sont sortis.

b) Problème de synchronisation rencontrés

L'unique problème de synchronisation rencontré dans cette partie était que tous les séances pouvaient débiter à n'importe quel moment. sans se soucier si les autres avaient commencé ou pas.

c) Solutions

Ajout de variables de contrôle et revérification du programme à l'aide de plusieurs tests et le **scheduling** pour les séances.

d) Traces d'exécutions

Stand de popcorn ouverte

Le client 1 a achete un billet pour la Seance 1

Nbrs de places restantes pour la seance: Seance 1: 2

Le client 2 a achete un billet pour la Seance 2

Nbrs de places restantes pour la seance: Seance 2: 1

Debut de la seance Seance 1

Entree client 1

Fin de la seance Seance 1

Sortie client 1

Debut de la seance Seance 2

Entree client 2

Fin de la seance Seance 2

Sortie client 2

2 Un cinémas avec un stand de popcorn**a) Contexte de simulation**

A ce niveau, il y a un stand de popcorn gratuit dans le cinéma et chaque client peut ou non passer par le stand de popcorn juste après avoir eu son ticket et/ou avant d'entrer dans la salle de projection

b) Problème de synchronisation rencontrés

Nous avons ajouté un autre Thread serveur et un Monitor popcorn, mais nous avons eu quelques soucis, le serveur ne servait qu'un seul client puis s'arrêtait, et par manque de temps nous avons pas su comment régler ce problème.

Conclusion

L'objectif de ce TP était de mettre en œuvre le fonctionnement d'un cinémas pour bien voir les problèmes de synchronisation qui pouvait y avoir.

Ce travail a été fait sur plusieurs parties, avec dans chaque partie l'ajout d'une nouvelle fonctionnalité à notre système (cinéma), permettant de découvrir et de résoudre à chaque fois de nouveaux problèmes de synchronisation.

Ce TP nous a appris beaucoup choses sur les Multi-Thread , important pour mètre en œuvre des applications qui prennent plusieurs acteurs.

Nous nous sommes ainsi familiariser à synchroniser des processus qui partagent une ressource en lecture/écriture.