Université du Québec à Chicoutimi
Département d'informatique et de mathématique

8INF957 – Programmation objet avancée : TP2

Professeur : Hamid Mcheick	Groupe : deux étudiant(e)s au maximum
Session : Aut2021	Date de distribution : 30 septembre 2021
Pondération : 15 points	Date de remise : 09 novembre 2021

Objectifs

Le but de ce projet est de familiariser les étudiants avec les concepts OO avancés suivants :

- SOLID
- Cohésion et couplage
- Généricité, Héritage & Délégation
- Programmation concurrente (Multithreading)
- Diagramme de classes
- Gestion d'exceptions
- Robustesse de logiciels
- Expression Lambda, NIO
- Séparation de préoccupations, AOP (AspectJ), SOP (HyperJ), VOP
- Programmation orientée composants
- Programmation orientée Microservices

Vous devez sélectionner **deux questions** parmi les questions suivantes.

Question 1 (50%)

Pour ce travail, on donnera à manger aux Pigeons !! Le but est d'implémenter une simulation d'alimentation de pigeon dans un espace public. Le jeu se passe dans une fenêtre où les pigeons attendent la nourriture. L'utilisateur, alors, leur donne à manger en cliquant sur un emplacement dans la fenêtre.

Les pigeons suivent les règles suivantes :

i. Chaque pigeon est contrôlé par un thread
ii. Si rien ne se passe, les pigeons s'endorment et ne bougent pas
iii. En apercevant de la nourriture, un pigeon se déplace vers la nourriture la plus fraiche.
iv. Une nourriture fraiche touchée est mangée, donc elle doit disparaitre immédiatement de la scène.
v. Nos pigeons sont gâtés ; un pigeon qui touche une nourriture pas fraiche, il l'ignore.
vi. Même en l'absence de la nourriture, des fois les pigeons se font effrayer et ils se dispersent à des positions aléatoires. Intégrer ce mécanisme dont la probabilité d'occurrence change d'un tour à l'autre.

Les pigeons et la nourriture doivent être représentés graphiquement. Un simple cercle ou un objet 3D, c'est à vous de choisir. Le plus important est le multithreading et la structure des classes. En ce qui Concerne le multithreading, vous serez amené à :

i. S'assurer que les pigeons arrêtent de bouger le moment où il ne reste plus de nourriture.
ii. Aussi, s'assurer si plus qu'un pigeon touche la nourriture simultanément, uniquement un seul pourra la supprimer.
iii. Prendre en compte la nourriture dans le processus de dessin. Puisque les threads ne sont pas synchronisés, ceci permet d'ajouter de la nourriture même au moment du dessin de la scène. Il sera commode de parcourir une structure de nourriture et de faire un verrou pendant le processus de dessin.

Vous avez le choix en tout ce qui est la conception des pigeons, leurs comportements, leurs dimensions sur la fenêtre, vitesse, le nombre … votre code est censé être robuste, des exceptions non gérées coutent des points.

Livrables :

- Votre travail (code et pdf/word) en un fichier .zip
- Le fichier README qui contient vos noms (groupe) et les instructions pour jouer à votre jeu.
- Document word/pdf décrivant les différentes parties pour répondre à cette question.
- Notons que la gestion d'exceptions et de multithreading va être considérée dans ce partie.

Question 2 (50%)

L'objectif est de prendre contact avec la programmation orientée aspect. Vous avez à modifier un jeu d'échec. Cependant, en général aucune modification au code source n'est permise, mais vous aurez la flexibilité pour modifier le code si nécessaire. Vous devrez ajouter des aspects:

i. Implémenter la validation des déplacements avec des Aspects (indiqué dans le code). Cette validation peut être : a) le joueur déplace une pièce et non une case vide, b) cette pièce lui appartient, c) le mouvement est autorisé par la pièce en question, d) le mouvement final ne sort pas de l'échiquier (1<= xfinal, yfinal <= 8), e) la pièce ne mange pas une pièce qui appartient à ce même joueur, f) la pièce ne saute pas d'autres pièces, excepté pour le cavalier.
ii. Journaliser tous les coups joués dans un fichier (pas besoin de noter les prises, les promotions, etc.). Placer un coup par ligne.
iii. Le code fonctionne avec le code source original.

*NB : **le code était implémenté un peu à l'arrache pour le travail, vous avez une certaine flexibilité pour le modifier selon les besoins (faites vos choix).***

Livrables :

i. L'implantation avec les répertoires *src* et *bin* contenant respectivement les *.java*, les *.aj* et les *.class*. Normalement, cela devrait être les fichiers originaux auxquels les aspects ont été ajoutés.
ii. Un document qui explique comment vous implémentez ces aspects non-fonctionnels.
iii. Le fichier README qui contient vos noms inclus dans le fichier.zip.
iv. Votre projet en un fichier .zip

Question 3 (50%)

L'architecture microservice est une variante du style architectural de l'architecture orientée services (SOA) - qui structure une application comme un ensemble de services faiblement couplés. C'est une technique de développement logiciel permettant de réduire la redondance et le couplage entre les divers composants. Les microservices communiquent les uns avec les autres en utilisant des API indépendantes du langage de programmation.

The term "Microservice Architecture" has sprung up over the last few years to describe a particular way of designing software applications as suites of independently deployable services: Martin Fowler.

Le terme microservice est apparu en 2011 au cours d'ateliers d'architecture (1), bien qu'il réutilise un grand nombre de principes largement employé par les systèmes d'information des grandes entreprises, notamment les concepts de l'architecture orientée service (SOA) (2).

1 - « Microservices », sur martinfowler.com (consulté le 20 février 2019), Martin Fowler : https://martinfowler.com/articles/microservices.html

2- « SOA versus microservices : quelles différences ? », (consulté le 20 février 2019): https://nexworld.fr/soa-versus-microservices-quelles-differences/

a) Expliquez ce style architectural Microservices en utilisant des figures
b) Appliquez ce style sur une étude de cas (application de ce style dans un programme complet).

Livrables :

- Le fichier README qui contient vos noms (groupe).
- Document word/pdf décrivant les différentes parties de microservices
    - Introduction, architecture microservices, avantages et inconvénients
    - Application de Miscroservice pour concevoir une architecture d'une application

Question 4 (50%)

Considérons un espace public sur lequel on a un graphe de n nœuds, distribués aléatoirement. Vous devez développer un jeu sérieux pour acheminer un message d'un nœud de départ A à un nœud d'arrivé B. Un point C, entre A et B, peut gagner un point si le message m_i arrive à B (m_i passe par C avant d'arriver à B). Nous souhaitons envoyer m messages entre A et B. Le nœud qui aura plus de points, est le gagnant. Vous pouvez appliquer la généricité, l'expression Lambda, etc.

a) Développer et tester un jeu pour acheminer m messages de A à B.
Hypothèses :

- Supposons que le graphe est connexe
- Supposons que la distance entre deux nœuds P et Q dans le graphe doit être plus petite ou égale à une variable d pour que le message m_i passe de P à Q.

- Supposons que vous utilisez la distance euclidienne pour calculer la distance entre deux nœuds dans le graphe P et Q.

À faire :

- Concevoir ou utiliser un algorithme pour trouver, pour un nœud donné n_i, la liste des nœuds connectés à n_i selon la distance (plus petite ou égale à une variable d).
- Concevoir un algorithme pour sélectionner le prochain nœud selon la plus petite distance.
- Concevoir un algorithme pour acheminer m messages de A à B.

b) Bonus (2 points): Supposons que la distance entre P et Q est plus grande que d. Une idée pourrait être utile pour résoudre ce problème est : P peut se déplacer vers Q ou Q peut se déplacer vers P. Dans ce contexte, comment peut-on acheminer un message m_i du nœud A (le départ) au nœud B (la destination). Donnez un algorithme pour résoudre ce problème.

Livrables :

i. La conception et l''implantation de jeu *.java ou .cpp ou Python ou autre*.
ii. Un document qui explique comment vous avez implémenter jeu.
iii. Le fichier README qui contient vos noms inclus dans le fichier.zip.
iv. Votre travail en un fichier .zip