François Monteil

Flavien Verschaeve

Alexandre Bodart

Compte rendu java

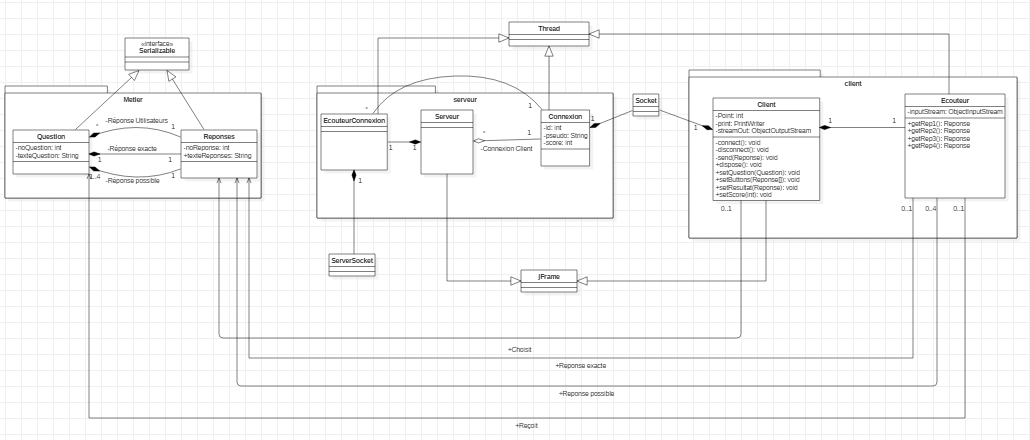


Table des matières

[Objectifs 5](#_Toc5123628)

[Répartition des tâches 6](#_Toc5123629)

[Diagramme de classe UML 7](#_Toc5123630)

[ 7](#_Toc5123631)

[Client 8](#_Toc5123632)

[Serveur 9](#_Toc5123633)

[Conclusion et amélioration possible 10](#_Toc5123634)

# Objectifs

|  |  |
| --- | --- |
| Objectif | Réalisé |
| Créer un modèle objet pertinent |  |
| Gérer des processus légers (Thread) |  |
| Échanger entre processus via des sockets |  |
| Stocker/lire des données dans une base de données. |  |
| Echange d’objets ( et non plus de chaînes de caractères) entre applications échangeant via Socket. |  |

# Répartition des tâches

|  |  |
| --- | --- |
| Objectif | Responsable |
| Créer un modèle objet pertinent | François |
| Gérer des processus légers (Thread) | François/Flavien/Alexandre |
| Échanger entre processus via des sockets | François/Alexandre |
| Stocker/lire des données dans une base de données. | Flavien/Alexandre |
| Echange d’objets ( et non plus de chaînes de caractères) entre applications échangeant via Socket. | François/Alexandre |

Plus précisément :

Alexandre est la personne qui est chargé de s’occuper de la partie serveur ce qui inclût :

1. Créer un socket d’écoute.
2. Accepter les connexions client.
3. Communiquer avec les clients.
4. Faire le traitement des questions et des réponses.
5. IHM serveur.

François est en charge de la partie client :

1. Créer un socket de connexion.
2. Se connecter au serveur.
3. Communiquer avec le serveur.
4. Traitement des questions et réponses reçues du serveur.
5. IHM Client.

Flavien est en charge de la partie BDD :

1. Créer un répertoire de question et de leurs réponses sous forme d’une bdd.
2. Permettre la connexion à la bdd depuis l’application.
3. Instancier des objets question et reponse à l’aide de la BDD.

Outil de mise en commun du travail : GitHub <https://github.com/francoismonteil/Kahoot>

# Diagramme de classe UML

# 

# Client

Le but du client est de se connecter à un serveur, ici kahoot.

La première chose que fait le client une fois connecté au serveur c’est demander au joueur son pseudo afin de pouvoir le fournir au serveur.

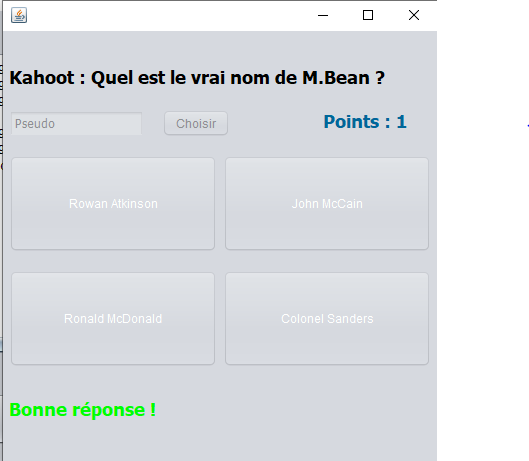
Une fois que le serveur à le pseudo du client il peut lui envoyer une question, le client reçoit la question entière :

* Texte de la question
* Les réponses possibles (sous forme d’objets Reponse)
* La réponse exacte

Nous avons fait en sorte que le client voit la question et n’ai la possibilité de répondre que par les réponses possibles sous forme de bouton, ici 4.

Quand le client clique sur l’une des réponses, celle-ci est automatiquement envoyer au serveur, nous vérifions si sa réponse correspond à la réponse exacte et on affiche un message en conséquence.

Puis nous attendons que le serveur nous envois notre score pour pouvoir l’afficher (et le stocker).



# Serveur

# Conclusion et amélioration possible

Nous avons remarqué que si un client se connecte en cours de partie alors l’intégrité du jeu est remise en cause, il faudrait alors faire une vérification des participants à chaque question OU bloqué la connexion des clients une fois la partie lancer.

De plus il faudrait bloquer le démarrage de partie tant que x joueurs ne se sont pas connectés et qu’ils n’ont pas donné leurs pseudos, car sinon la première réponse qu’ils fourniront deviendra leurs pseudos et ils ne pourront plus répondre à la question.

Nous n’avons pas automatisé l’envoi des questions, elle nécessite que nous appuyons sur début de partie à chaque fin de questions, il faudra alors une fois le bouton cliquer faire une boucle qui envoie les questions dès que tous les clients ont répondu.