

30 JANVIER 2019



Rendu 1



À l'attention de M. Teyssié et M. Torguet

DOSSIER JAVA

PROJET MULTI-USER DUNGEON

BOURAKKEDI – LAPEYRADE – OLIVIER

M1 STRI INGE UPSSITECH

118 route de Narbonne, 31062 Toulouse

Sommaire

Présentation	2
Étape 1 :	2
Introduction.....	2
Conception	3
Diagramme de cas d'utilisation	3
Diagramme de classes participantes	4
Diagrammes de séquence	5
Résultats et commentaires.....	8

Présentation

Le but de ce projet est de créer un jeu de rôles massivement multijoueur de type MUD (Multi-User Dungeon) en Java avec les Sockets TCP, UDP et/ou RMI. Nous avons personnellement choisi d'utiliser seulement RMI.

Par souci de granularité, la réalisation de ce projet fut divisée en 5 étapes majeures que l'organisation de ce dossier reprendra. Nous nous attacherons pour chacune d'entre elles à les présenter avant de fournir la conception et le code source leur correspondant.

Étape 1 :

Introduction

La première étape est la base du programme, elle comprend la mise en place d'un serveur de Donjon et de discussion auquel plusieurs joueurs pourront se connecter en simultané. Le donjon est un damier composé de pièces dans lequel les joueurs pourront se déplacer dans les pièces nord, sud, est et ouest.

Après s'être connectés en entrant le nom de leur personnage, les joueurs arrivent dans la pièce de départ et les joueurs seront prévenus si d'autres personnages sont dans la même pièce ou arrivent dans celle-ci. Nous avons également ajouté une notification quand un joueur quitte la pièce. Les joueurs pourront communiquer entre eux par le biais du serveur de discussion.

Notre implémentation se fera donc entièrement sur le principe de RMI (Remote Method Invocation) où les joueurs (clients) se connecteront aux différents serveurs.

Conception

Diagramme de cas d'utilisation

Voici le diagramme de cas d'utilisation de l'étape 1 :

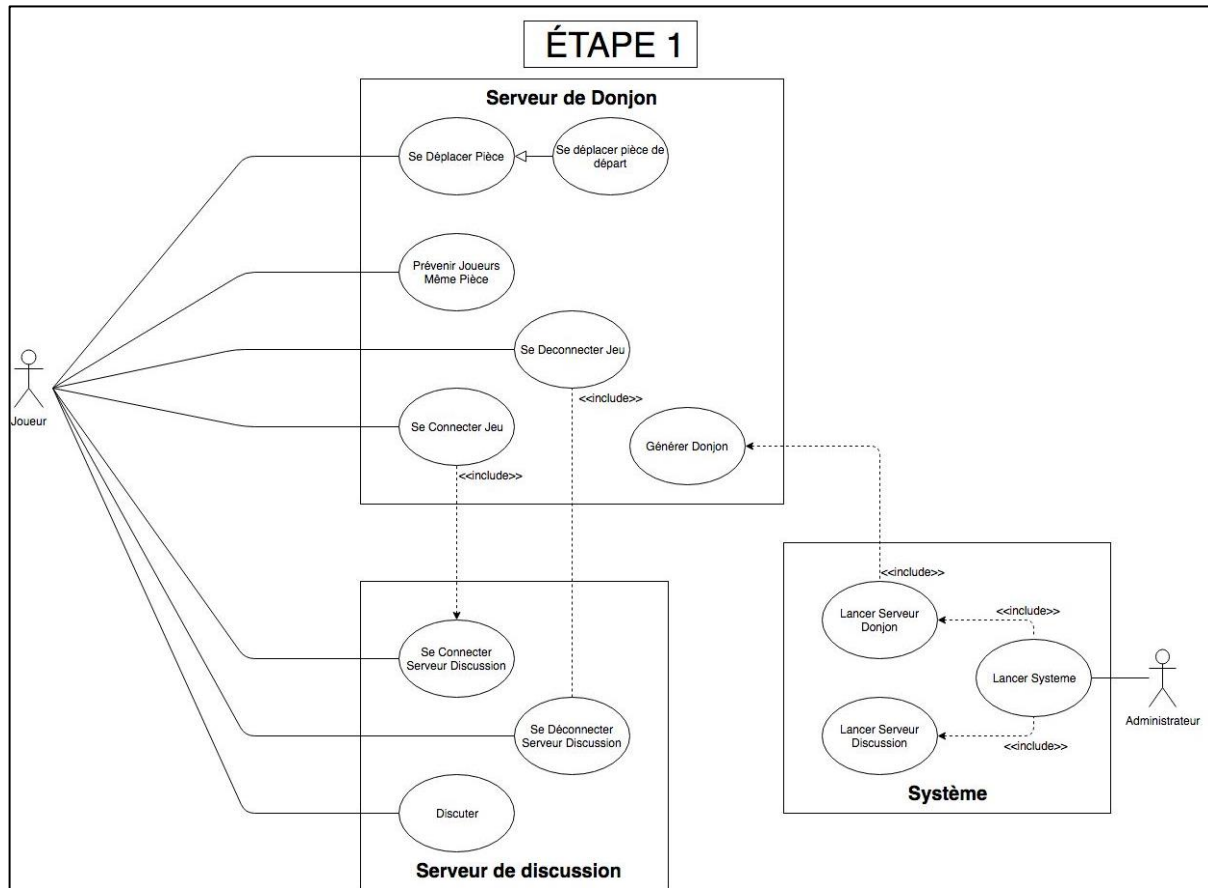


Figure 1: Diagramme de cas d'utilisation de l'étape 1

Nous détaillerons en diagramme de séquences les cas d'utilisation suivant :

- Cas « Se déplacer pièce »
- Cas « Se connecter jeu »
- Cas « Discuter »
- Cas « Lancer Système »

Ces derniers reprennent en effet les différentes fonctionnalités principales de la première étape.

Diagramme de classes participantes

Voici le diagramme de classes participantes de notre étape 1 illustrant notre implémentation de notre Diagramme de cas d'utilisation. On y retrouve les différentes classes et leurs relations respectives. On remarque la présence de l'objet « Remote » nécessaire à l'implémentation de RMI.

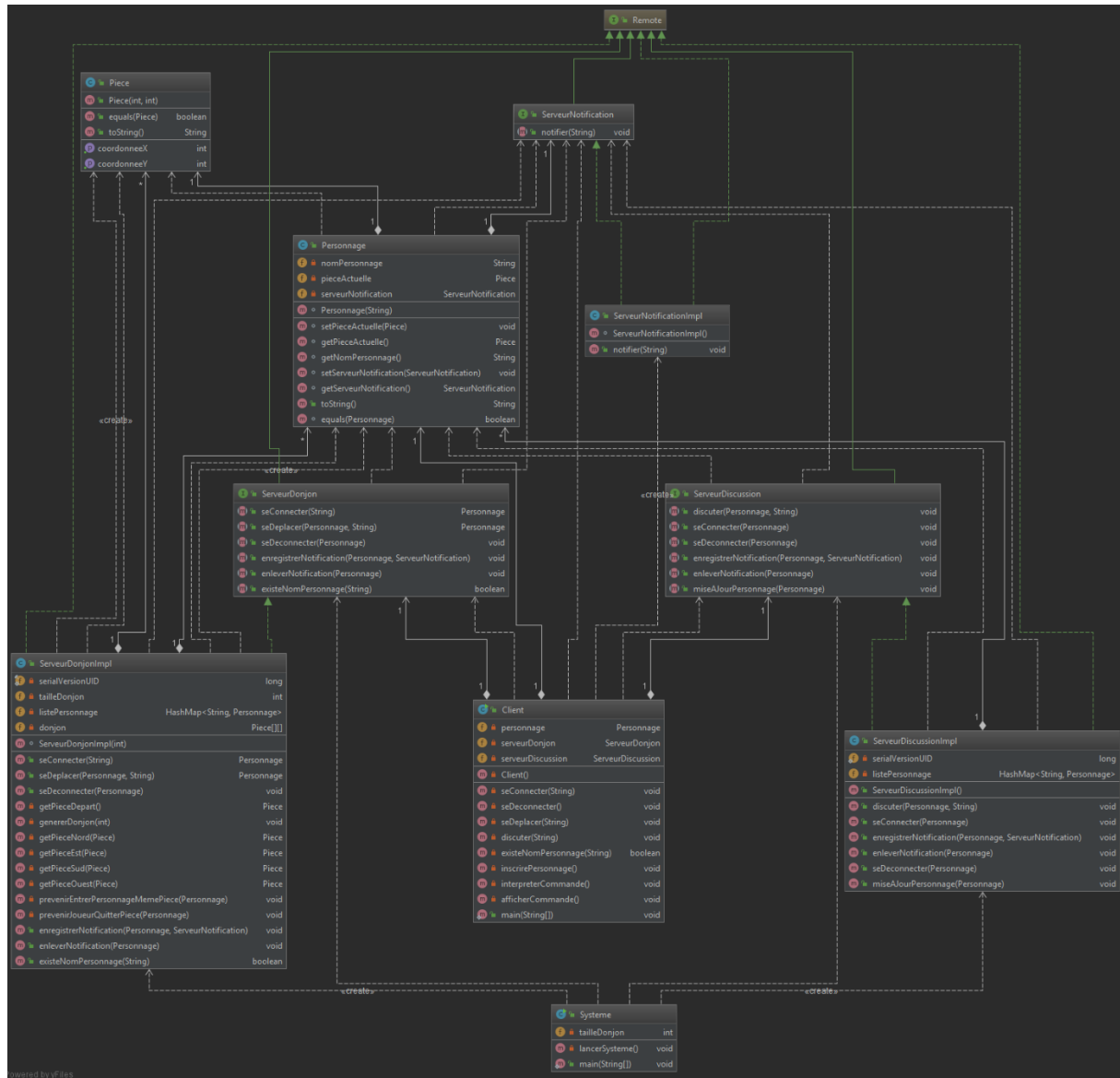


Figure 2: Diagramme de classes participantes de l'étape 1

Diagrammes de séquence

Cas « Se déplacer pièce »

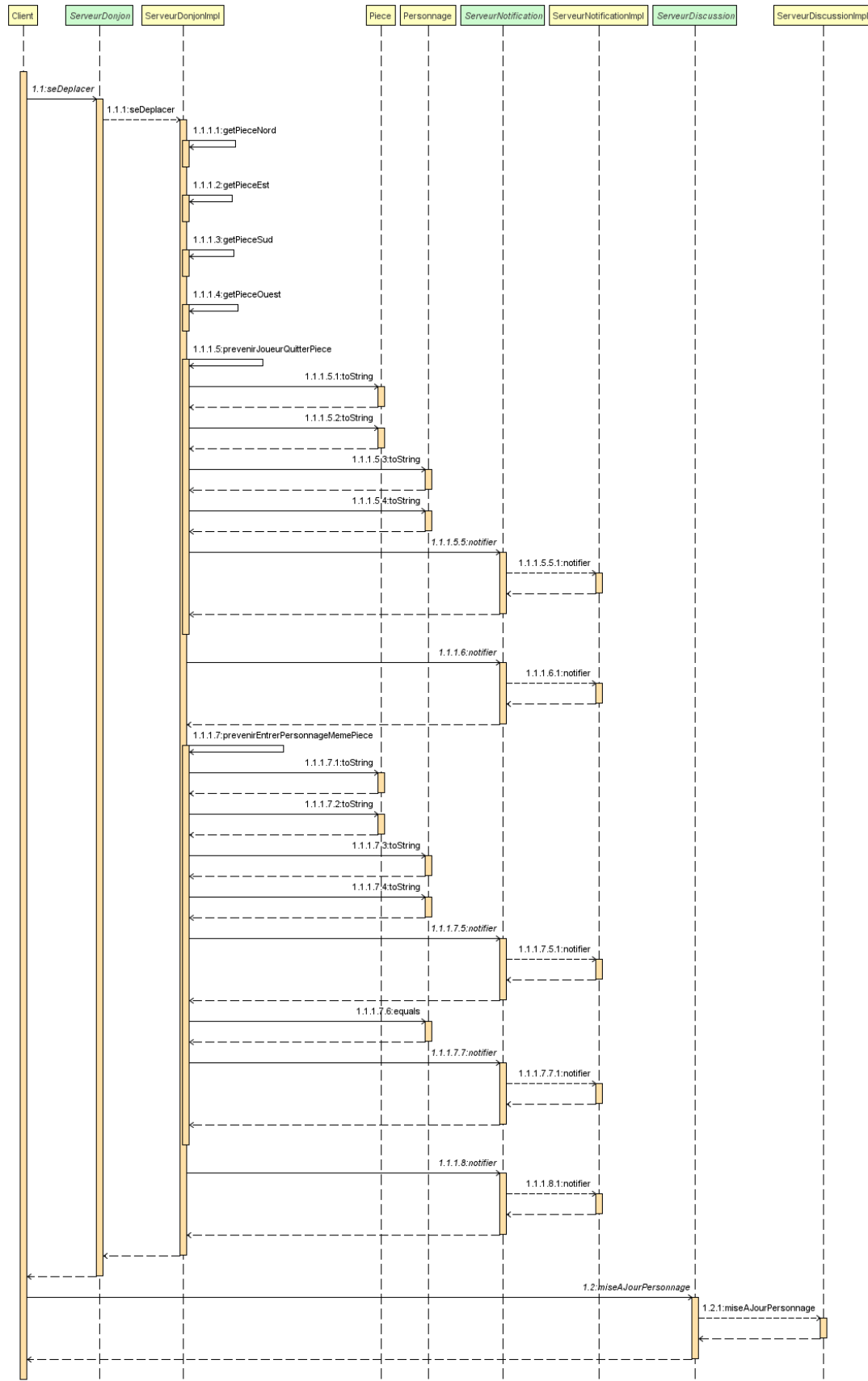


Figure 3: Diagramme de séquence du cas "Se déplacer"

Cas « Se connecter jeu »

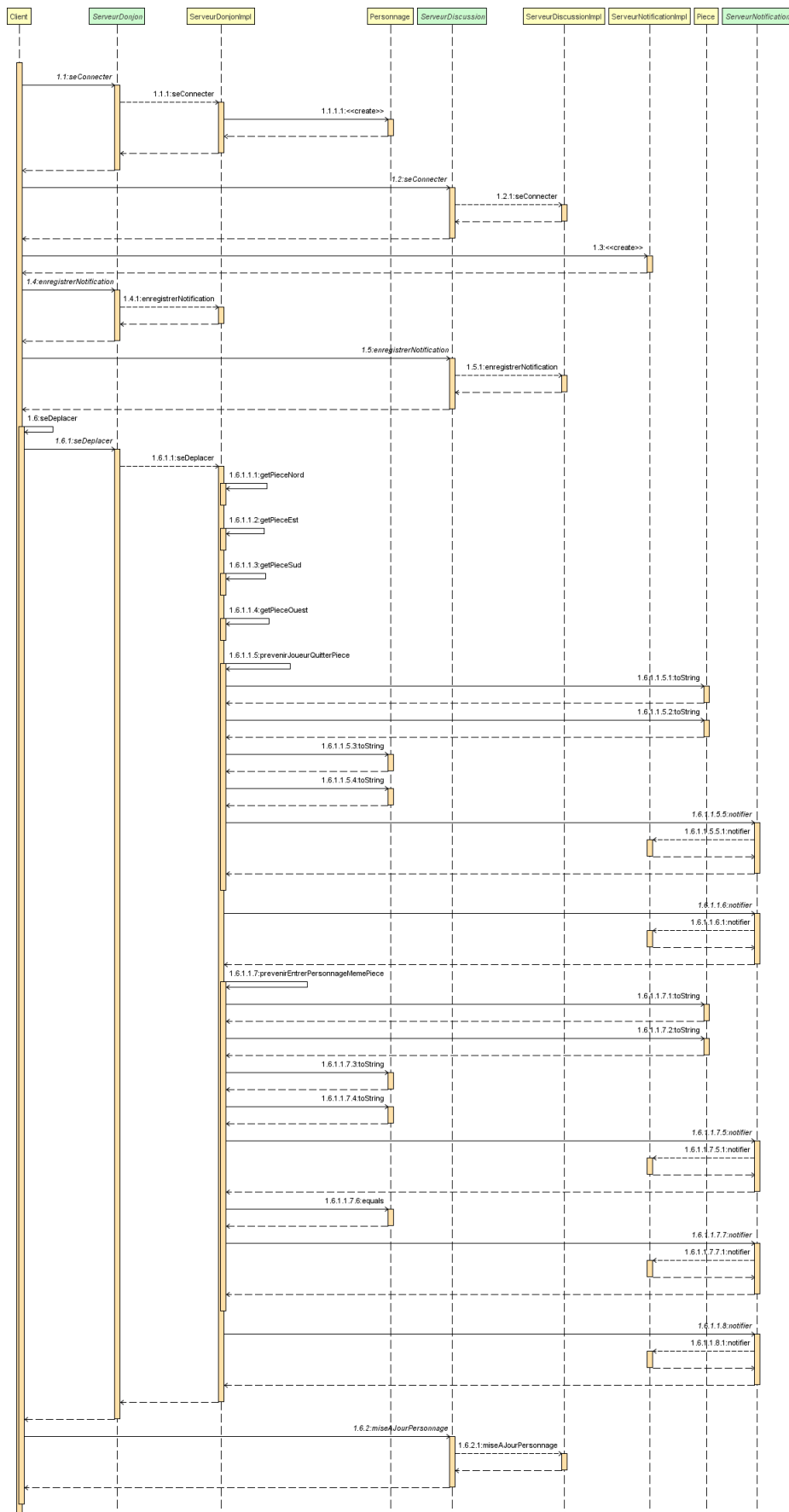


Figure 4: Diagramme de séquence du cas "Se connecter"

Cas « Discuter »

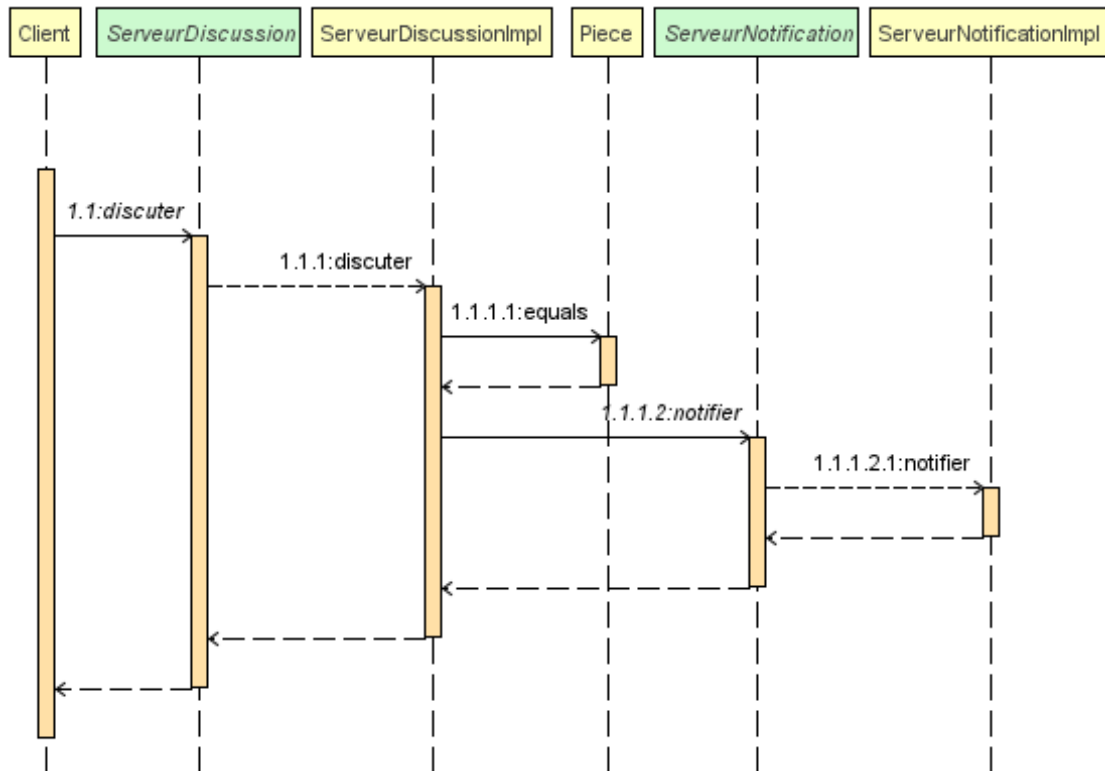


Figure 5: Diagramme de séquence du cas "Discuter"

Cas « Lancer Système »

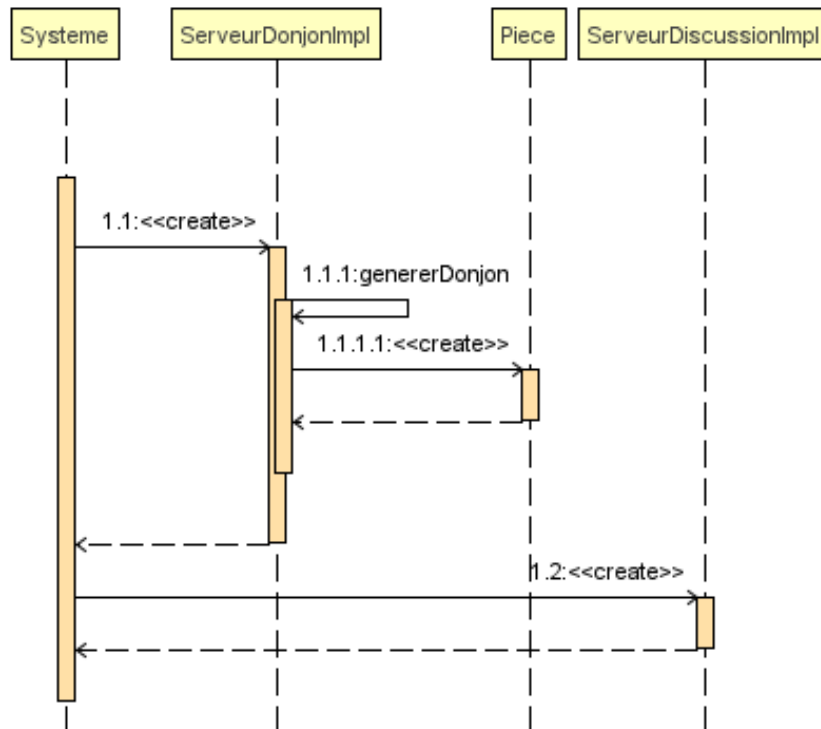


Figure 6: Diagramme de séquence du cas "Lancer Système"

Résultats et commentaires

Sur la Figure 7, nous pouvons observer que nous pouvons connecter plusieurs personnages en simultané. De plus, lorsqu'un personnage se déconnecte, il peut être reconnecté avec le même nom.

```

C:\Program Files\Java\jdk-11.0.1\bin\java.exe
Le serveur donjon est démarré.
Le serveur discussion est démarré.
Connexion de Personnage[Jean].
Connexion de Personnage[Paul].
Déconnexion de Personnage[Paul]
Déconnexion de Personnage[Jean]
Connexion de Personnage[Jean].

```

Figure 7: Capture d'une exécution du Système de l'étape 1

Sur la Figure 8, nous pouvons voir que le personnage « Jean » est connecté dans la pièce de départ « 0,0 ». Quand, un second personnage « Paul » se connecte dans la même pièce de départ, le premier est bien prévenu. Les deux personnages peuvent alors communiquer entre eux, Jean reçoit bien le message de Paul. De même quand, ce dernier quitte la pièce, le personnage restant est notifié.

Après avoir changé de pièce, Paul ne peut plus recevoir de messages envoyés par Jean comme nous pouvons le vérifier Figure 9.

Concernant les directions, la pièce de départ se trouvant à l'extrémité sud-ouest du damier, le personnage ne peut aller au Sud ni à l'Est depuis la pièce de départ, et un message l'en notifie bien. Si le personnage va dans une direction où une pièce existe, le serveur le notifie de la pièce où il arrive et si d'autres joueurs sont présents.

Enfin si le joueur entre la commande « quitter », il est bien déconnecté du serveur et le processus est bien terminé. On peut alors se reconnecter avec son nom, et seulement alors.

```

C:\Program Files\Java\jdk-11.0.1\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2018.3\lib\idea_rt.jar=49906:C:\Program Files\JetBrains\IntelliJ
Entrez votre nom de personnage : Jean
Le personnage Jean vient de se connecter.
Vous arrivez dans la pièce Piece[0:0]
Il n'y a pas d'autre joueur dans la pièce.

Entrer 'N', 'E', 'S' ou 'O' pour vous déplacer ou ' ' pour communiquer avec d'autres joueurs ou 'quitter' pour vous déconnecter ou 'help' pour voir les instructions.

Paul est entré dans la pièce: Piece[0:0]
Main Paul
Jean: Salut Paul
Paul: Bonjour Jean
Paul a quitté la pièce.
Main
Jean: Paul?

Impossible d'aller dans cette direction.
Vous arrivez dans la pièce Piece[1:0]
Il n'y a pas d'autre joueur dans la pièce.
Vous arrivez dans la pièce Piece[0:0]
Il n'y a pas d'autre joueur dans la pièce.
quitter
Déconnexion.

Process finished with exit code 0

```

Figure 8: Capture d'une exécution d'un premier client de l'étape 1

Comme nous pouvons observer au début de la Figure 9, Jean étant connecté, on ne peut pas se connecter avec son nom.

On observe de plus sur cette dernière capture que pour communiquer avec d'autres personnages, le joueur doit bien précéder son message par « " ». Sinon, le client est notifié que la commande n'est pas reconnue.

Enfin, nous pouvons constater que lorsque le joueur entre la commande « help », les commandes possibles lui sont bien rappelées.

```

Systeme x Client x Client x
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2018.3\lib\idea_rt.jar=49916:C:\Program Files\JetBrains\IntelliJ
Entrez votre nom de personnage : Jean
Ce nom existe déjà.
Entrez votre nom de personnage : Paul
Le personnage Paul vient de se connecter.
Vous arrivez dans la piece Piece[0:0]
Il y a Jean dans la pièce.

Entrez 'N', 'E', 'S' ou 'O' pour vous déplacer ou '' pour communiquer avec d'autres joueurs ou 'quitter' pour vous déconnecter ou 'help' pour voir les instructions.

Jean: Salut Paul
Bonjour Jean
Cette commande n'est pas reconnue.
Bonjour Jean
Paul: Bonjour Jean

Vous arrivez dans la piece Piece[1:0]
Il n'y a pas d'autre joueur dans la pièce.
"Quelqu'un est ici ?"
Paul: Quelqu'un est ici ?
Impossible d'aller dans cette direction.
Vous arrivez dans la piece Piece[1:1]
Il n'y a pas d'autre joueur dans la pièce.
help

Entrez 'N', 'E', 'S' ou 'O' pour vous déplacer ou '' pour communiquer avec d'autres joueurs ou 'quitter' pour vous déconnecter ou 'help' pour voir les instructions.
quitter
Déconnexion.

Process finished with exit code 0

```

Figure 9: Capture d'une exécution d'un second client de l'étape 1

Figure 1: Diagramme de cas d'utilisation de l'étape 1	3
Figure 2: Diagramme de classes participantes de l'étape 1	4
Figure 3: Diagramme de séquence du cas "Se déplacer"	5
Figure 4: Diagramme de séquence du cas "Se connecter"	6
Figure 5: Diagramme de séquence du cas "Discuter"	7
Figure 6: Diagramme de séquence du cas "Lancer Système"	7
Figure 7: Capture d'une exécution du Système de l'étape 1.....	8
Figure 8: Capture d'une exécution d'un premier client de l'étape 1	8
Figure 9: Capture d'une exécution d'un second client de l'étape 1.....	9