

18 FEVRIER 2019



## Rendu 2



*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

## Table des matières

Présentation .....	2
Étape 1 : .....	2
Introduction étape 1 .....	2
Conception étape 1 .....	3
Diagramme de cas d'utilisation étape 1 .....	3
Diagramme de classes participantes 1 .....	4
Diagrammes de séquence 1 .....	5
Résultats et commentaires 1 .....	8
Étape 2 : .....	9
Introduction étape 2 .....	9
Conception étape 2 .....	10
Diagramme de cas d'utilisation 2 .....	10
Diagramme de classes participantes 2 .....	11
Diagrammes de séquence 2 .....	12
Résultats et commentaires 2 .....	13
Table des illustrations .....	15

## 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 présenter leurs caractéristiques avant de fournir leur conception, et un ensemble de scénarios commentés. Le code source leur correspondant sera fourni à part.

## Étape 1 :

### Introduction étape 1

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 simultanée. 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. Les joueurs seront prévenus si d'autres personnages sont dans la même pièce ou arrivent dans celle-ci. Pour les prévenir, le donjon enverra un message via le serveur de notification propre à RMI.

Nous avons également ajouté une notification quand un joueur quitte la pièce. Les joueurs pourront communiquer entre eux, tant qu'ils sont dans la même pièce, par le biais du serveur de discussion qui sera dédié à cela.

Enfin l'ensemble des commandes exécutables leur sera présenté lors de la connexion ou en entrant la commande « help » s'ils désirent les consulter à nouveau.

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

## Conception étape 1

### Diagramme de cas d'utilisation étape 1

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

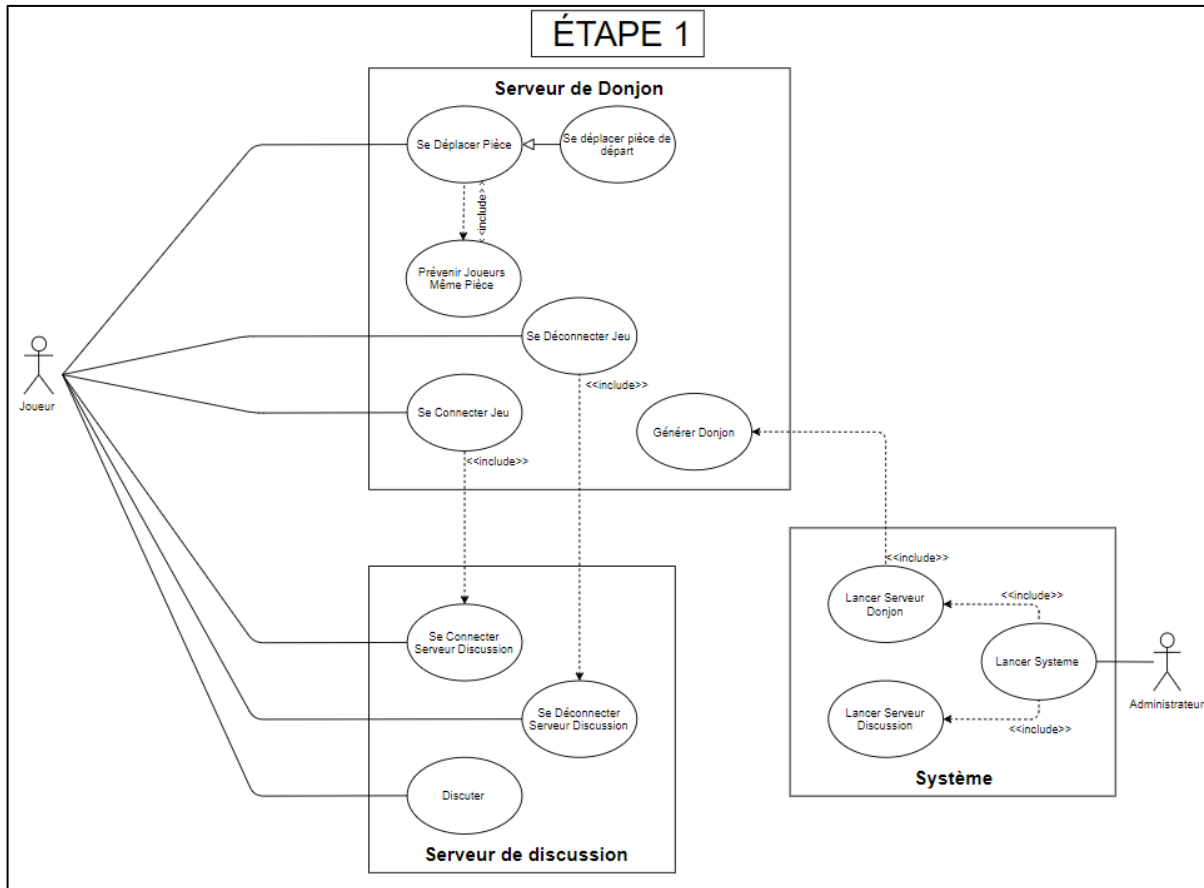


Figure 1: Diagramme de cas d'utilisation de l'étape 1

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 1

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 :

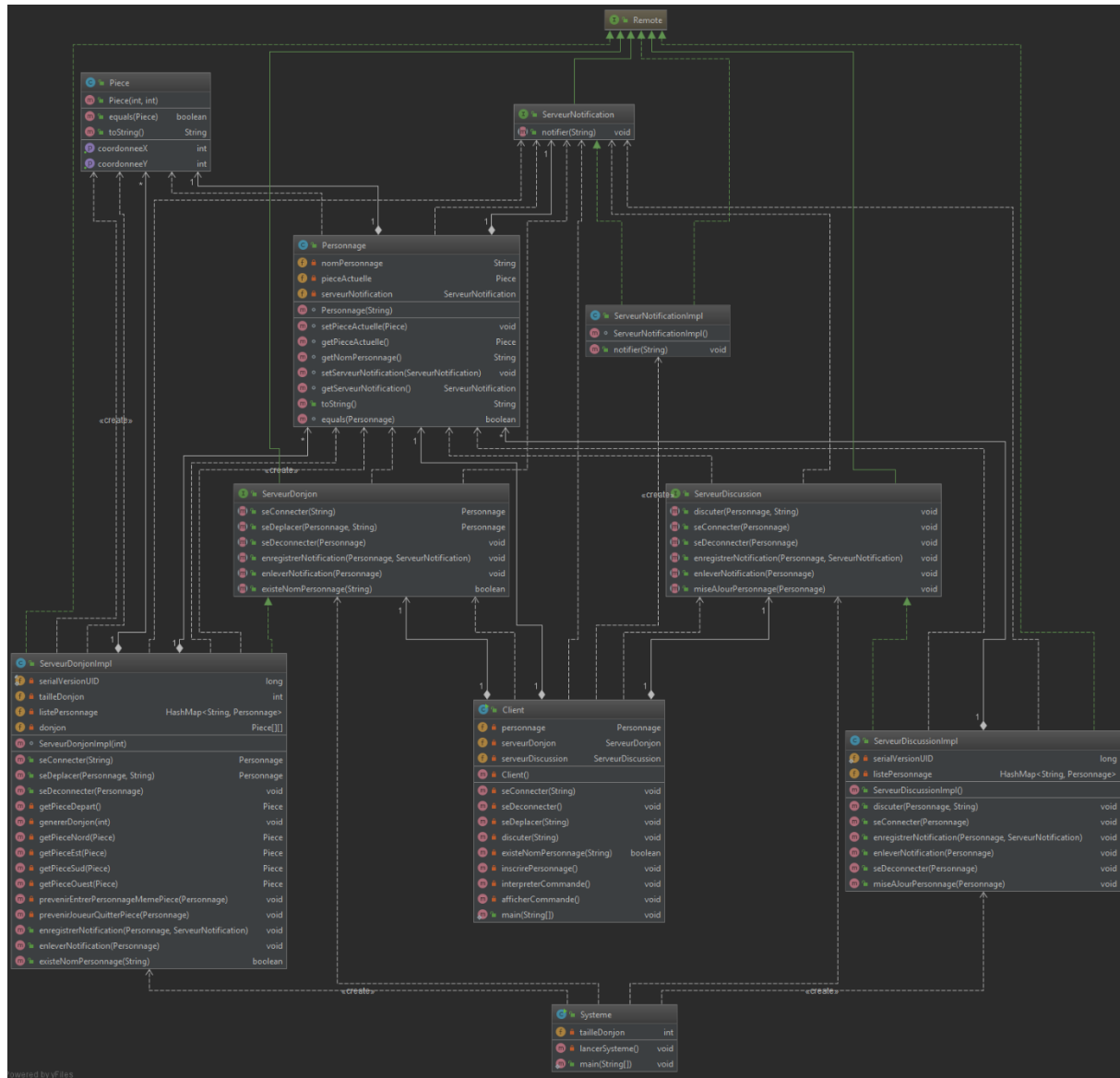


Figure 2: Diagramme de classes participantes de l'étape 1

## Diagrammes de séquence 1

### Cas « Se déplacer pièce »

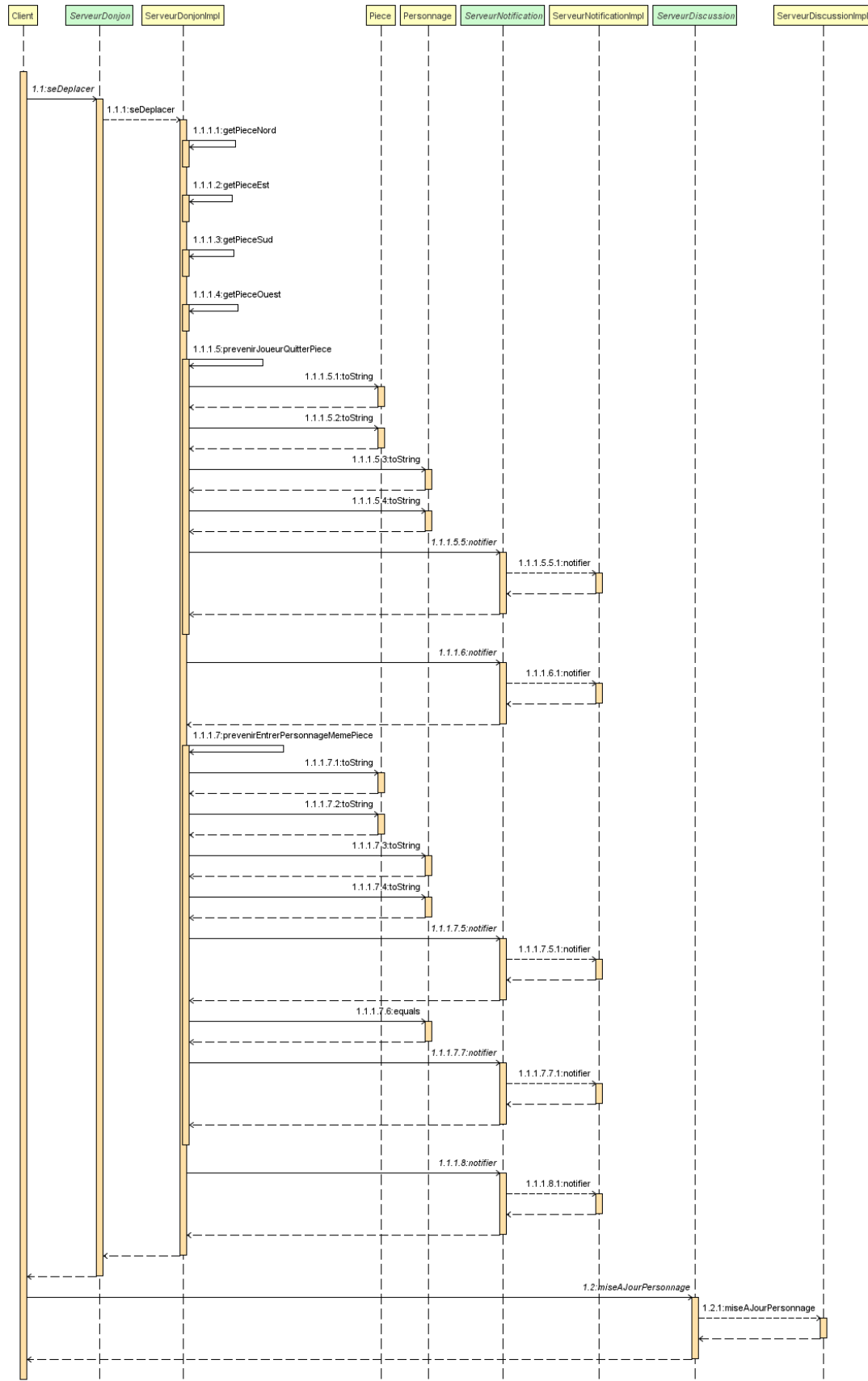


Figure 3: Diagramme de séquence du cas "Se déplacer"

## Cas « Se connecter jeu »

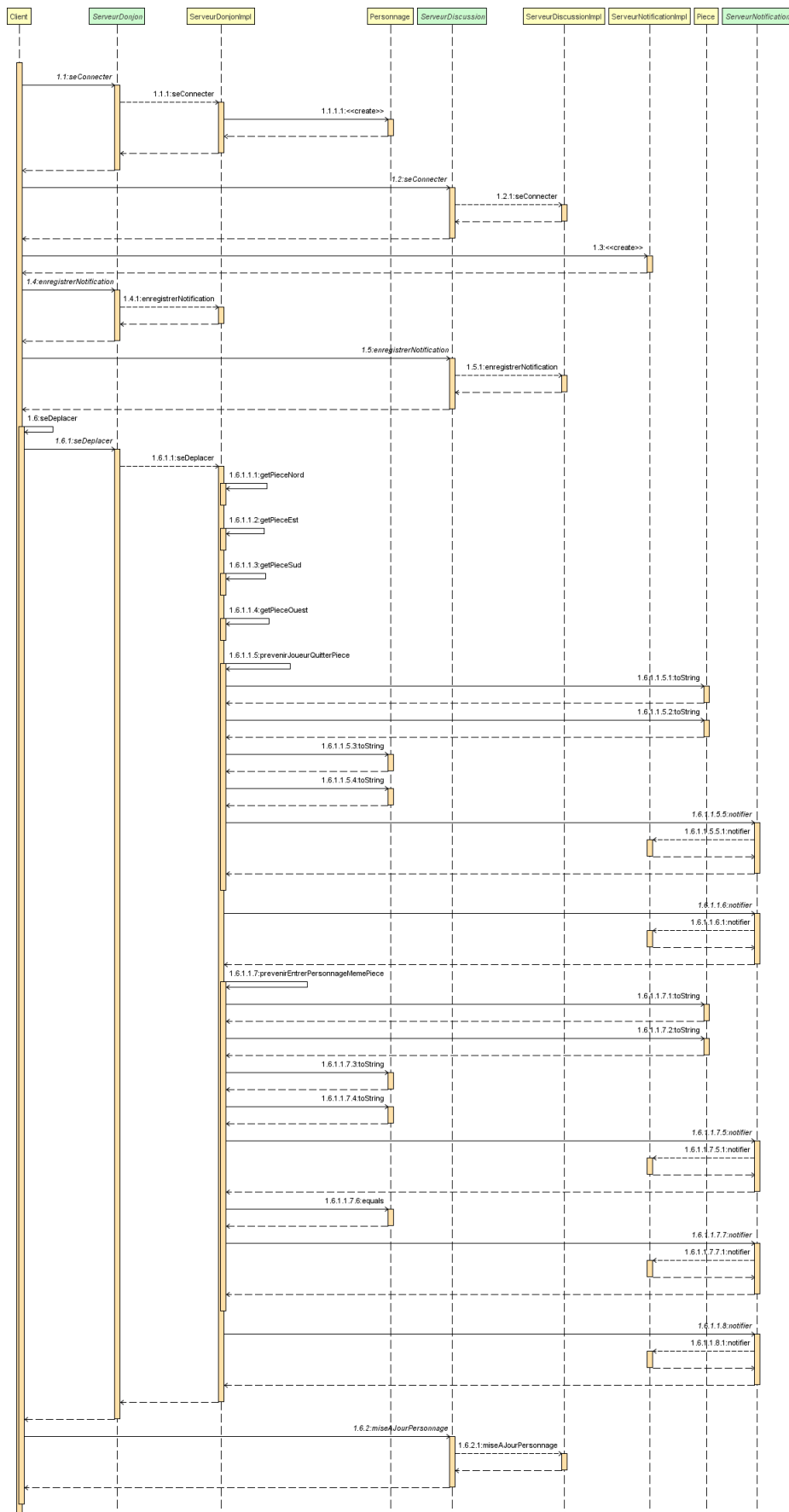


Figure 4: Diagramme de séquence du cas "Se connecter"

Cas « Discuter »

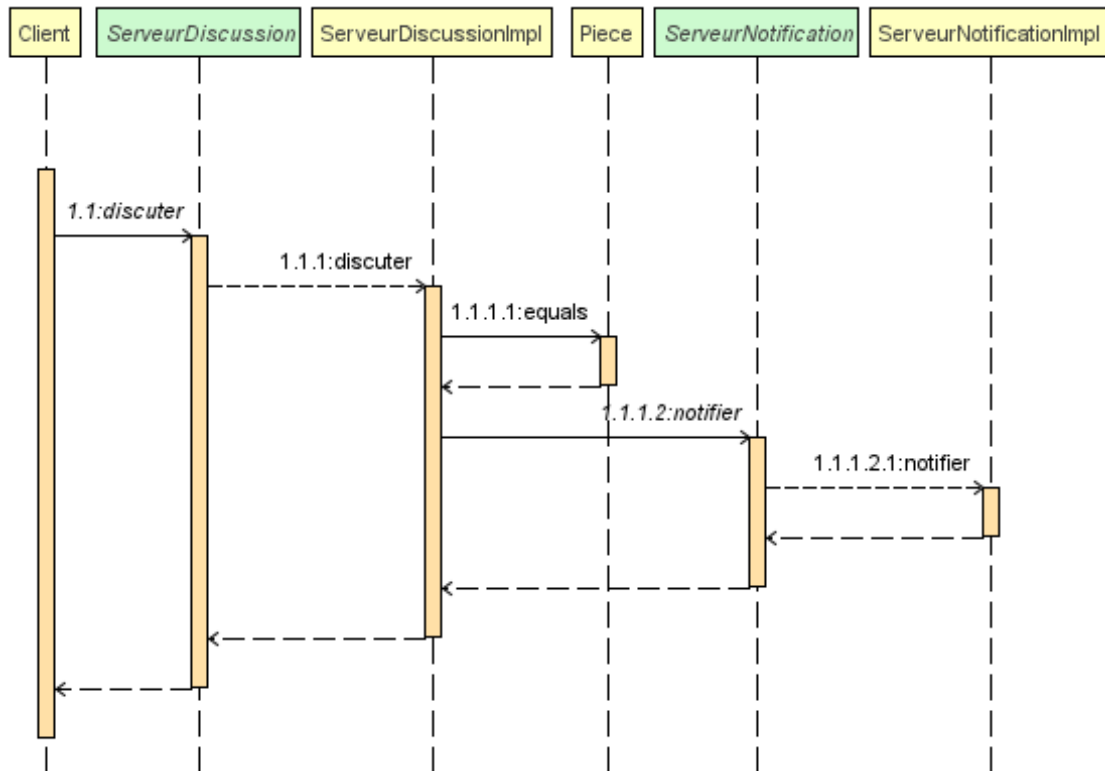


Figure 5: Diagramme de séquence du cas "Discuter"

Cas « Lancer Système »

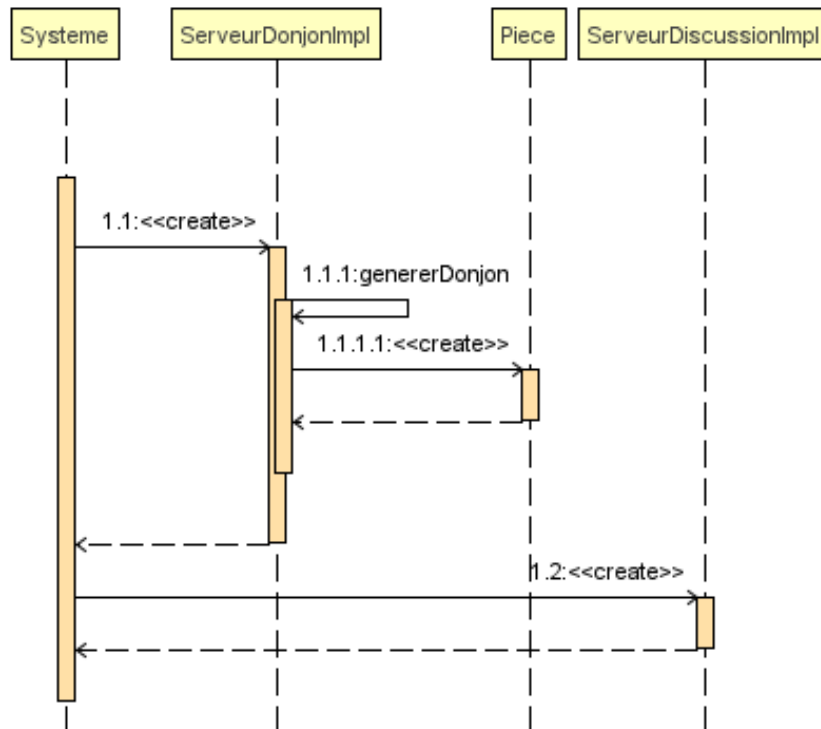


Figure 6: Diagramme de séquence du cas "Lancer Système"



## Résultats et commentaires 1

Sur la Figure 7, nous pouvons observer que nous pouvons connecter plusieurs personnages en simultanée. 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'ouest 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
Entrer votre nom de personnage : Jean
Le personnage Jean vient de se connecter.
Vous arrivez dans la piece 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]
Jean: Salut Paul
Paul: Bonjour Jean
Paul a quitté la pièce.
Jean: Paul?
Impossible d'aller dans cette direction.
Vous arrivez dans la piece Piece[1:0]
Il n'y a pas d'autre joueur dans la pièce.
Vous arrivez dans la piece Piece[0:0]
Il n'y a pas d'autre joueur dans la pièce.
quitter
Déconnexion.
Process finished with exit code 0

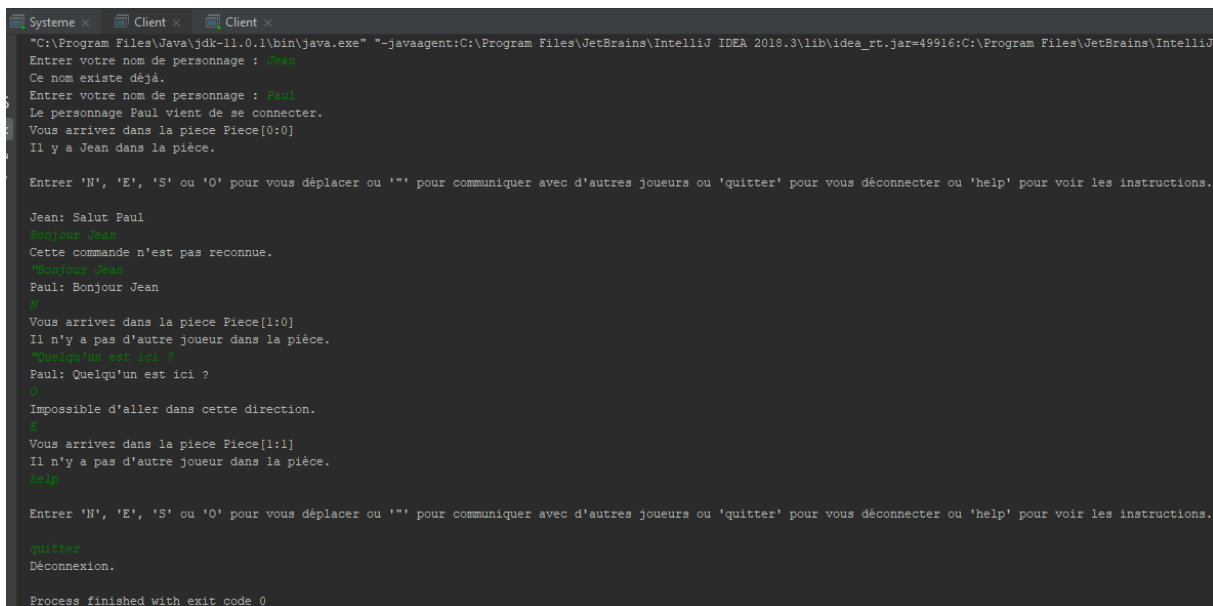
```

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.
Quitter
Déconnexion.

Process finished with exit code 0

```

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

## Étape 2 :

### Introduction étape 2

La deuxième étape introduit les monstres et les combats. Nous représenterons les premiers en créant une classe mère « ÊtresVivants » dont hériteront les personnages déjà présents et les monstres. Les combats et les monstres seront créés lors de l'entrée d'un personnage dans une pièce. Ce personnage aura alors le choix entre fuir le combat ou le continuer.

Les combats se déroulent sous forme de tours, lors de chacun, un participant perd aléatoirement un point de vie. Sachant que les monstres commencent avec 5 points de vie et les personnages 10. Si un des participants arrive à 0 point de vie, il meurt et disparaît donc du donjon.

Lorsqu'un monstre ou un personnage remporte un combat, il gagne un point de vie supplémentaire. S'il n'y a pas d'autre personnage dans la pièce, l'ensemble des êtres encore vivants regagnent leurs points de vie maximum.

Comme pour la discussion de l'étape 1, les combats se dérouleront sur un serveur dédié, à part, que nous appellerons « ServeurCombat ».

## Conception étape 2

### Diagramme de cas d'utilisation 2

Voici le diagramme de cas d'utilisation repris en ajoutant les éléments de l'étape 2 :

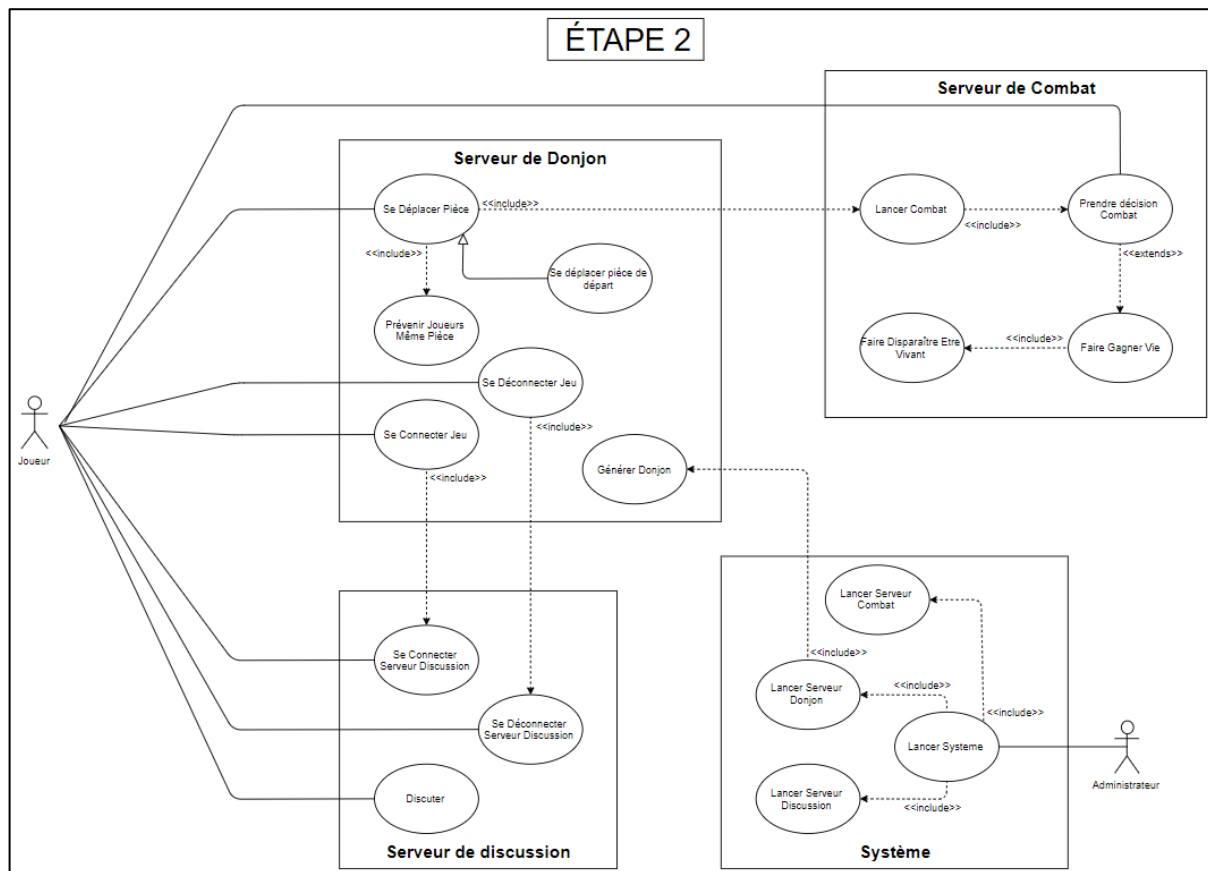


Figure 10: Diagramme de cas d'utilisation de l'étape 2

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

- Cas « Lancer Combat »
- Cas « Faire Gagner Vie »

Ces derniers reprennent en effet les nouveautés et modifications principales de la deuxième étape.

## Diagramme de classes participantes 2

La Figure 11 représente le diagramme de classes participantes partiel de notre étape 2 illustrant notre implémentation de notre Diagramme de cas d'utilisation. On y retrouve, par souci de lisibilité, seulement les principales classes que nous avons ajoutées et modifiées ainsi que leurs relations respectives. Les principales sont la classe « EtreVivant » dont la classe « Personnage » et « Monstre » hériteront. Les classes « ServeurCombat » et « ServeurCombatImpl » sont également ajoutées selon l'architecture RMI. Enfin nous avons introduit une classe « CombatMonstre » s'occupant des combats entre Montre et Personnage. La précédente classe « ServeurDonjon » a été modifiée en « Donjon » et comprend maintenant la liste unique des personnages, monstres et combats :

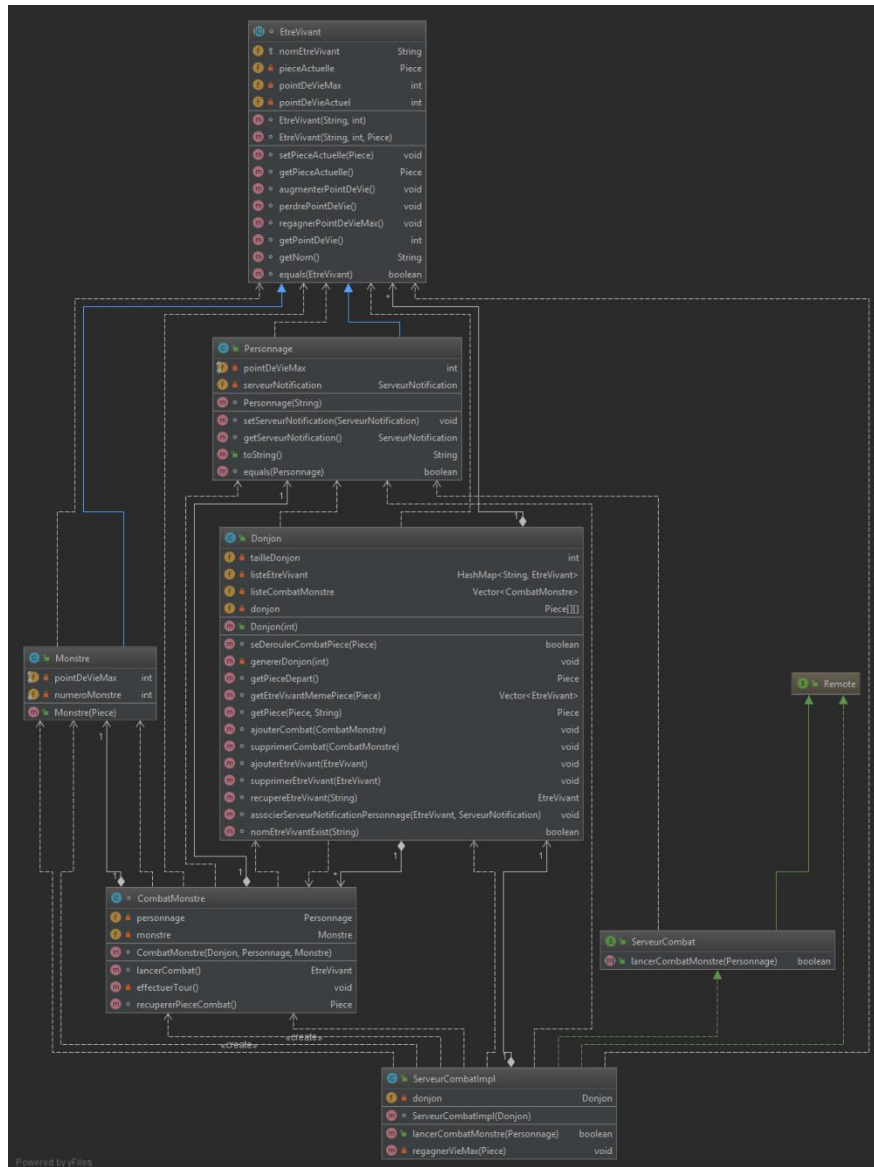


Figure 11: Diagramme de classes participantes partiel de l'étape 2

## Diagrammes de séquence 2

### Cas « Lancer Combat »

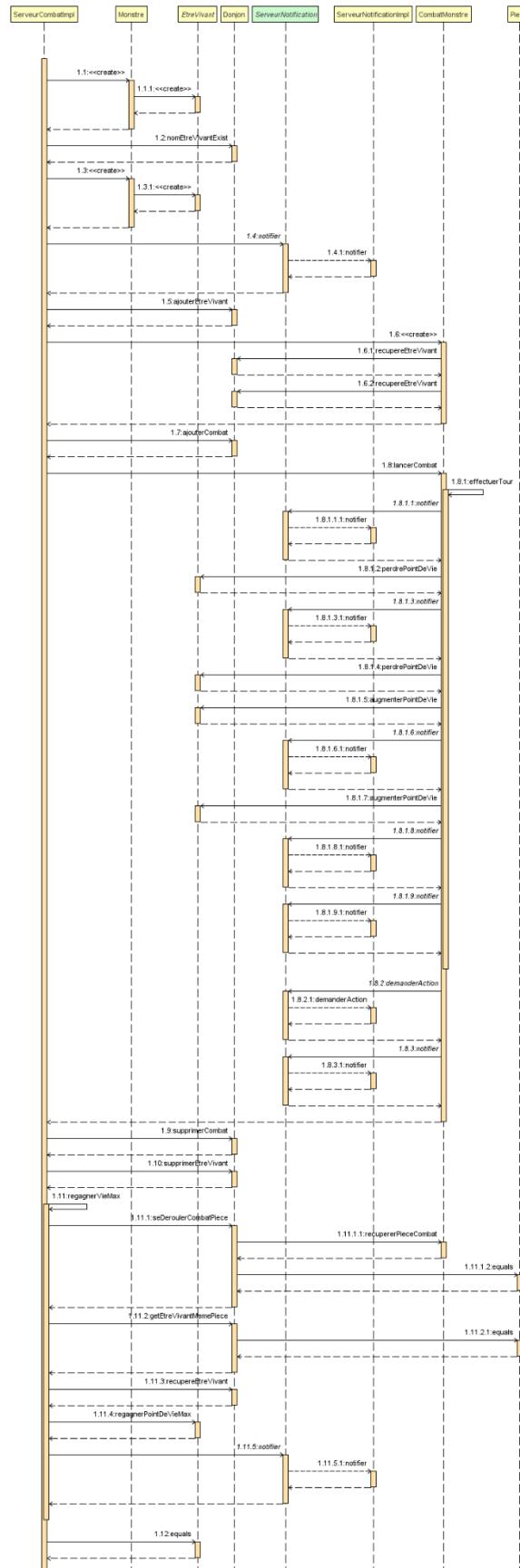


Figure 12: Diagramme de séquence du cas "Lancer Combat"

## Cas « Faire Gagner Vie »

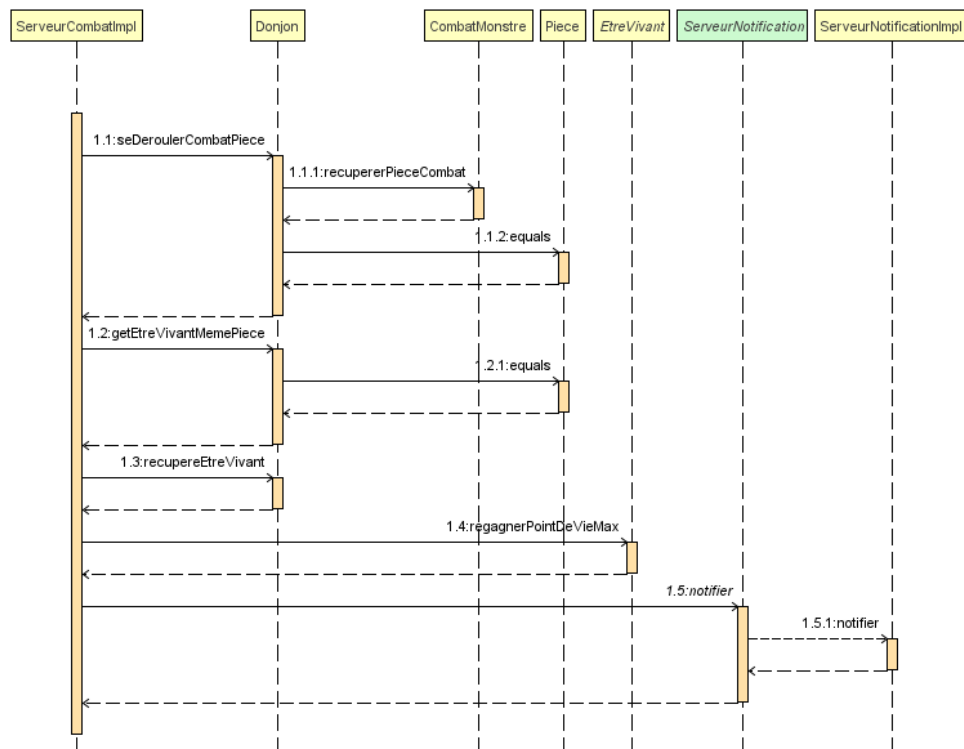


Figure 13: Diagramme de séquence du cas "Faire Gagner Vie"

## Résultats et commentaires 2

Lors du lancement du système, nous pouvons sur la Figure 14 que le serveur de combat est lancer en plus des deux autres serveurs de l'étape précédente.

```

Systeme x Client x Client x Client x
"C:\Program Files\Java\jdk-11.0.1\bin\java.exe"
Le serveur donjon est démarré.
Le serveur discussion est démarré.
Le serveur combat est démarré.
    
```

Figure 14: Capture de l'exécution du système

Nous allons à présent présenter sur la Figure 15 et la Figure 16 un scénario présentant un combat. Sur la première illustration, un personnage nommé « Jean » se connecte puis entre la commande "N" pour se déplacer à la pièce au nord. Un monstre apparaît et l'attaque alors, puisqu'il est entré dans une nouvelle pièce.

On peut ensuite voir le déroulement du combat, à chaque tour, le joueur appuie sur entrée pour continuer le combat et un des participants perd un point de vie. Sur ce scénario, on observe que c'est le personnage qui perd, il est donc déconnecté et il disparaît du donjon alors que le monstre reste.

```

Entrer 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, '.' pour communiquer, 'quitter' pour
vous déconnecter, 'L' pour afficher les êtres dans la pièce ou 'help' pour voir les
commandes.

Vous arrivez dans la pièce : Piece[1:0]
Il n'y a pas d'autre joueur dans la pièce.
Vous êtes attaqué par Monstre0
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 9 pdv. Le monstre a 5 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 8 pdv. Le monstre a 5 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 7 pdv. Le monstre a 5 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Paul est entré dans la pièce: Piece[1:0]
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 6 pdv. Le monstre a 5 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 5 pdv. Le monstre a 5 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 4 pdv. Le monstre a 5 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 3 pdv. Le monstre a 5 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 2 pdv. Le monstre a 5 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Vous attaquez Monstre0 perd 1 point de vie.
Il vous reste 2 pdv. Le monstre a 4 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Monstre0 vous attaque, vous perdez 1 point de vie.
Il vous reste 1 pdv. Le monstre a 4 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Monstre0 vous attaque, vous perdez 1 point de vie.
Vous mourez... bye bye.
Process finished with exit code 0

```

Figure 15: Exécution d'un premier client lors de l'étape 2

De même lorsqu'un second personnage nommé « Paul » apparaît dans la même pièce, le premier personnage est notifié, comme on peut l'observer sur la capture, sans interrompre le combat.

Sur la seconde capture, nous observons la même scène du point de vue de Paul. Celui-ci arrive dans la pièce où Jean se bat avec un monstre. Il voit donc les deux êtres en arrivant. Il est alors logiquement attaqué par un monstre.

Après un tour où le monstre perd un point de vie, il choisit de fuir. Les commandes disponibles lui sont alors rappelées. Il liste alors les êtres vivants dans la pièce en entrant la commande "L" et voit qu'il y a toujours Jean et un monstre qui n'ont pas fini de combattre et un second monstre qu'il a lui-même fui.

Lors de la fin du combat de Jean, il n'y a plus de combat dans la pièce, les êtres vivants encore présents regagnent donc tous leur point de vie maximum. Enfin lorsque Paul liste à nouveau les membres présents dans la pièce, il n'y a plus Jean, mais le monstre qu'il combattait est bien toujours présent.

```

Entrer votre nom de personnage : Paul
Le personnage Paul 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, '.' pour communiquer, 'quitter' pour
vous déconnecter, 'L' pour afficher les êtres dans la pièce ou 'help' pour voir les
commandes.

Vous arrivez dans la pièce : Piece[1:0]
Il y a Jean Monstre0 dans la pièce.
Vous êtes attaqué par Monstre1
Vous attaquez Monstre1 perd 1 point de vie.
Il vous reste 10 pdv. Le monstre a 4 pdv.
Entrez "fuir" pour quitter le combat ou n'importe quelle autre commande pour continuer :
Vous avez fui le combat. Il vous reste 10 point de vie.

Entrer 'N', 'E', 'S' ou 'O' pour vous déplacer, '.' pour communiquer, 'quitter' pour
vous déconnecter, 'L' pour afficher les êtres dans la pièce ou 'help' pour voir les
commandes.

Il y a Jean Monstre0 Monstre1 dans la pièce.
Il n'y a plus de combat en cours, vous regagnez vos points de vie maximum. Vous avez 10
pdv.

Il y a Monstre0 Monstre1 dans la pièce.

```

Figure 16: Exécution d'un second client lors de l'étape 2

## Table des illustrations

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
Figure 10: Diagramme de cas d'utilisation de l'étape 2 .....	10
Figure 11: Diagramme de classes participantes partiel de l'étape 2.....	11
Figure 12: Diagramme de séquence du cas "Lancer Combat" .....	12
Figure 13: Diagramme de séquence du cas "Faire Gagner Vie" .....	13
Figure 14: Capture de l'exécution du système.....	13
Figure 15: Exécution d'un premier client lors de l'étape 2 .....	14
Figure 16: Exécution d'un second client lors de l'étape 2 .....	14