Table des matières

[1 Introduction (Yassin) 5](#_Toc421960118)

[2 Analyse 6](#_Toc421960119)

[2.1 Règles du jeu 6](#_Toc421960120)

[2.1.1 Préparatifs du jeu 6](#_Toc421960121)

[2.1.2 Distribution du matériel 6](#_Toc421960122)

[2.1.3 Placement de départ des pions des joueurs 6](#_Toc421960123)

[2.1.4 Déroulement du jeu 6](#_Toc421960124)

[2.1.5 Fin du jeu 7](#_Toc421960125)

[2.1.6 Règles complémentaires 7](#_Toc421960126)

[2.2 Partage des responsabilités entre serveur et client 8](#_Toc421960127)

[2.2.1 Client 8](#_Toc421960128)

[2.2.2 Serveur 9](#_Toc421960129)

[2.3 Diagrammes d’activité (RAPHAEL) 10](#_Toc421960130)

[2.4 Cas d’utilisation 11](#_Toc421960131)

[2.4.1 Diagramme général de contexte 11](#_Toc421960132)

[2.4.2 Description des acteurs 11](#_Toc421960133)

[2.4.3 Scénarios 11](#_Toc421960134)

[2.5 Base de données 20](#_Toc421960135)

[2.5.1 Objectifs 20](#_Toc421960136)

[2.5.2 Modèle entité-association (VANESSA) 20](#_Toc421960137)

[2.5.3 Contraintes d’intégrité 20](#_Toc421960138)

[2.5.4 Commentaires 21](#_Toc421960139)

[2.6 Modèle de domaine 21](#_Toc421960140)

[2.6.1 Modèle de domaine côté serveur 21](#_Toc421960141)

[2.6.2 Modèle de domaine côté client 23](#_Toc421960142)

[3 Conception 24](#_Toc421960143)

[3.1 Protocole d’échange entre le client et le serveur (RAPHAEL) 24](#_Toc421960144)

[3.2 Serveur (RAPHAEL) 24](#_Toc421960145)

[3.2.1 Architecture 24](#_Toc421960146)

[3.2.2 Diagramme de classes 24](#_Toc421960147)

[3.2.3 Package XYZ 24](#_Toc421960148)

[3.3 Client (YASSIN, VANESSA) 24](#_Toc421960149)

[3.3.1 Architecture(YASSIN) 24](#_Toc421960150)

[3.3.2 Diagramme de classes(YASSIN OU VANESSA) 24](#_Toc421960151)

[3.3.3 Description(YASSIN, VANESSA) 24](#_Toc421960152)

[3.3.4 Package XYZ 24](#_Toc421960153)

[3.4 Base de données (VANESSA) 24](#_Toc421960154)

[3.4.1 Modèle relationnel 24](#_Toc421960155)

[3.4.2 Script 25](#_Toc421960156)

[3.5 Structures Json (YASSIN) 26](#_Toc421960157)

[3.5.1 XXX 26](#_Toc421960158)

[3.6 Règles de définition de compte utilisateur 26](#_Toc421960159)

[3.6.1 Définition de nom d’utilisateur 26](#_Toc421960160)

[3.6.2 Définition de mot de passe 26](#_Toc421960161)

[4 Implémentation 27](#_Toc421960162)

[4.1 Technologies (Vanessa) 27](#_Toc421960163)

[4.2 Client(VANESSA) 27](#_Toc421960164)

[4.3 Serveur(Yassin) 27](#_Toc421960165)

[4.4 Problèmes rencontrés (Raphaël) 27](#_Toc421960166)

[5 Gestion du projet 28](#_Toc421960167)

[5.1 Rôle des participants 28](#_Toc421960168)

[5.1.1 Guillaume Bruge 28](#_Toc421960169)

[5.1.2 Yassin Kammoun (YASSIN) 28](#_Toc421960170)

[5.1.3 Michelle Meguep (VANESSA) 28](#_Toc421960171)

[5.1.4 Raphäel Racine (RAPHAEL) 28](#_Toc421960172)

[5.2 Plan d’itérations initial 28](#_Toc421960173)

[5.2.1 Itération 1 : 22.04.2015 - 29.04.2015 28](#_Toc421960174)

[5.2.2 Itération 2 : 29.04.2015 - 06.05.2015 29](#_Toc421960175)

[5.2.3 Itération 3 : 06.05.2015 - 13.05.2015 29](#_Toc421960176)

[5.2.4 Itération 4 : 13.05.2015 - 20.05.2015 30](#_Toc421960177)

[5.2.5 Itération 5 : 20.05.2015 - 27.05.2015 31](#_Toc421960178)

[5.2.6 Itération 6 : 27.05.2015 - 03.06.2015 31](#_Toc421960179)

[5.2.7 Itération 7 : 03.06.2015 - 10.06.2015 32](#_Toc421960180)

[5.2.8 Itération 8 : 10.06.2015 - 17.06.2015 32](#_Toc421960181)

[5.3 Plan d’itérations finale 33](#_Toc421960182)

[5.3.1 Itération 1 : 22.04.2015 - 29.04.2015 33](#_Toc421960183)

[5.3.2 Itération 2 : 29.04.2015 - 06.05.2015 34](#_Toc421960184)

[5.3.3 Itération 3 : 06.05.2015 - 13.05.2015 35](#_Toc421960185)

[5.3.4 Itération 4 : 13.05.2015 - 20.05.2015 37](#_Toc421960186)

[5.3.5 Itération 5 : 20.05.2015 - 27.05.2015 38](#_Toc421960187)

[5.3.6 Itération 6 : 27.05.2015 - 03.06.2015 41](#_Toc421960188)

[5.3.7 Itération 7 : 03.06.2015 - 10.06.2015 42](#_Toc421960189)

[5.3.8 Itération 8 : 10.06.2015 - 17.06.2015 45](#_Toc421960190)

[5.4 Suivi de projet 48](#_Toc421960191)

[5.4.1 Itération 1 : 22.04.2015 - 29.04.2015 48](#_Toc421960192)

[5.4.2 Itération 2 : 29.04.2015 - 06.05.2015 49](#_Toc421960193)

[5.4.3 Itération 3 : 06.05.2015 - 13.05.2015 51](#_Toc421960194)

[5.4.4 Itération 4 : 13.05.2015 - 20.05.2015 53](#_Toc421960195)

[5.4.5 Itération 5 : 20.05.2015 - 27.05.2015 54](#_Toc421960196)

[5.4.6 Itération 6 : 27.05.2015 - 03.06.2015 56](#_Toc421960197)

[5.4.7 Itération 7 : 03.06.2015 - 10.06.2015 57](#_Toc421960198)

[5.4.8 Itération 8 : 10.06.2015 - 17.06.2015 (TOUT LE MONDE) 59](#_Toc421960199)

[5.5 Stratégie de tests (RAPHAEL) 60](#_Toc421960200)

[5.6 Stratégie d’intégration du code de chaque participant (VANESSA) 60](#_Toc421960201)

[6 Etat des lieux 61](#_Toc421960202)

[6.1 Ce qui fonctionne (RAPHAËL) 61](#_Toc421960203)

[6.2 Ce qu’il resterait à développer (RAPHAËL) 61](#_Toc421960204)

[7 Auto-critique 62](#_Toc421960205)

[7.1 Solution technique(YASSIN) 62](#_Toc421960206)

[7.2 Gestion de projet(YASSIN) 62](#_Toc421960207)

[7.3 Planification(YASSIN) 62](#_Toc421960208)

[7.4 Améliorations(YASSIN) 62](#_Toc421960209)

[7.5 Critiques personnelles(TOUT LE MONDE) 62](#_Toc421960210)

[7.5.1 Guillaume Bruge 62](#_Toc421960211)

[7.5.2 Yassin Kammoun(YASSIN) 62](#_Toc421960212)

[7.5.3 Michelle Meguep(VANESSA) 62](#_Toc421960213)

[7.5.4 Raphaël Racine(RAPHAEL) 62](#_Toc421960214)

[8 Conclusion (YASSIN) 63](#_Toc421960215)

[9 Annexes 64](#_Toc421960216)

[9.1 Installation(Vanessa) 64](#_Toc421960217)

[9.2 Utilisation(VANESSA) 64](#_Toc421960218)

**Table des figures**

[Figure 1 : Diagramme général de contexte 11](#_Toc421958143)

[Figure 2 : Modèle de domaine côté serveur 22](file:///C:\Users\Yassin\Desktop\GEN_Scotland_Yard_Rapport.docx#_Toc421958144)

[Figure 3 : Modèle de domaine côté client 23](file:///C:\Users\Yassin\Desktop\GEN_Scotland_Yard_Rapport.docx#_Toc421958145)

# Introduction (Yassin)

# Analyse

Cette section présente la phase d’analyse du projet. Il s’agit dans un premier temps de présenter le jeu et ses règles. Dans un second temps, le fonctionnement global de l’application est présenté au moyen de cas d’utilisations. Des données persistantes allant être stockées dans le cadre du projet, une partie de cette section portera sur une première ébauche d’une base de données. Celle-ci sera suivie et conclut par le modèle de domaine respectif des applicatifs client-serveur.

## Règles du jeu

Cette partie se veut être une présentation détaillée des règles du jeu Scotland Yard allant faire l’objet d’une implémentation dans l’application client-serveur.

### Préparatifs du jeu

Les joueurs décident de celui qui endossera le rôle de Mister X. Il leur est fort conseiller de choisir de préférence un participant ayant les nerfs solides. Les joueurs restants constitueront l’équipe des détectives.

### Distribution du matériel

Les préparatifs du jeu seront suivis par la distribution du « matériel » du jeu. Ainsi, chaque type de joueur se verra distribuer d’une certaine quantité de ressources allant être utilisées durant la partie :

* Mister X se verra attribuer de :
* Le pion incolore.
* Le tableau de parcours.
* 4 tickets de taxi.
* 3 tickets de bus.
* 3 tickets de métro.
* Les 2 cartes « coup double ».
* Autant de tickets noirs qu’il y a de détectives participant au jeu.
* Chaque détective se verra attribuer de :
  + Un pion de couleur.
  + 10 tickets de taxi.
  + 8 tickets de bus.
  + 4 tickets de métro.

### Placement de départ des pions des joueurs

Afin de déterminer les points de départ des joueurs, les cartes de départ sont mélangées. Chaque joueur, Mister X y compris, tire ensuite une carte de départ. Puis les détectives placent leurs pions sur les cases portant le numéro de la carte de départ obtenue.

Il est important de noter que seul Mister X s’abstient de tout placement de pion. Il conserve donc le secret le plus absolu quant à son point de départ.

### Déroulement du jeu

Mister X ouvre le bal, puis c’est aux détectives de jouer à tour de rôle, ce qui constitue un tour complet. Chacun doit payer sa course en remettant le ticket correspondant. Mister X inscrit ses déplacements sur le tableau de parcours. Quant aux détectives, ils font avancer leurs pions sur le plateau de jeu.

#### Progression du jeu

Chaque coup correspond à un trajet en taxi, en métro ou en bus d’un arrêt à un autre le long du tracé en couleur. La course doit être payée avec le ticket de couleur correspondant. Chaque point représente un arrêt indiquant le(s) moyen(s) de transport y faisant halte. Les points à plusieurs couleurs représentent les arrêts de plusieurs moyens de transport. Là-bas, les joueurs peuvent, au coup suivant, changer de moyen de transport.

#### Les coups de Mister X

Mister X joue généralement ses coups dans le secret le plus absolu. Pour cela, il note en cachette, dans la fenêtre adéquate du tableau de parcours, le numéro de la case sur laquelle il voudrait se rendre. Il cache le numéro sous le ticket correspondant. A présent, les détectives connaissent le moyen de transport emprunté par Mister X, mais ignorent sa destination.

#### Premier coup de Mister X

Au premier coup, Mister X choisit un arrêt relié à son point de départ. Il note le numéro de cet arrêt dans la première fenêtre de son tableau de parcours et le cache au moyen du ticket correspondant. La couleur de ce ticket doit en effet correspondre à la ligne de couleur reliant le point de départ à ce premier arrêt.

#### Les coups des détectives

Après Mister X, c’est aux détectives de jouer à tour de rôle, toujours dans le sens des aiguilles d’une montre. Chacun remet un ticket et place son pion sur l’arrêt suivant du moyen de transport choisi par lui. Mister X se voit remettre les tickets utilisés (autrement dit le détective doit donner son ticket à Mister X).

Il est important de noter que, pour chaque moyen de transport, les détectives ne disposent que d’un nombre limité de tickets. Dès qu’un détective a consommé sa provision de tickets, il ne peut plus emprunter le moyen de transport en question. Mister X doit avoir connaissance des tickets que chaque détective a en sa possession, afin qu’il sache à tout moment où il peut encore être traqué.

#### Apparition de Mister X

Mister X se manifeste à intervalles réguliers, plus précisément à son 3ème, 8ème, 13ème, et 18ème tour. Les endroits où il doit faire son apparition sont indiqués par une fenêtre plus grande sur le tableau de parcours. Il continue à enregistrer ses déplacements sur le tableau de parcours en cachant chaque fois le numéro au moyen du ticket correspondant. Puis il place son pion à l’arrêt où il est arrivé.

#### Coups spéciaux de Mister X

Les points suivants constituent les coups spéciaux pouvant être portés par Mister X :

* Coup double
  + Mister X peut, en un seul et même coup, parvenir deux arrêts plus loin en combinant astucieusement les moyens de transport. Il note les deux arrêts et dépose deux tickets. La carte « coup double » est mise de côté. Si le premier coup le conduit à une station d’apparition, il doit se manifester, mais disparaît dès le deuxième coup
* Tickets noirs
  + Mister X peut à tout moment remettre un ticket noir à la place d’un ticket normal, ce qui lui permet d’emprunter n’importe quel moyen de transport. Les détectives ignorent alors le moyen de transport utilisé par Mister X.

### Fin du jeu

Mister X doit se manifester au moment où un détective arrive, avec son pion, sur l’arrêt servant momentanément de repaire à Mister X. La victoire revient alors aux détectives, autrement dit à l’équipe. Si, au contraire, Mister X parvient à rester caché jusqu’au 24ème coup ou que les détectives ne peuvent plus progresser, c’est lui qui remporte la partie.

### Règles complémentaires

Les points suivants constituent des règles complémentaires ne devant pas être négligées :

* Il ne peut jamais y avoir deux pions de détective sur un seul et même arrêt.
* Il est possible de faire marche arrière au prochain coup, à condition bien sûr de posséder le ticket correspondant.
* Les détectives n’ont pas le droit de s’échanger des tickets entre eux. Ils doivent chacun se satisfaire de leur propre réserve.
* L’ordre du jeu des détectives doit toujours être le même (c’est-à-dire à tour de rôle dans le sens des aiguilles d’une montre). Dans certaines situations délicates, le respect de cet ordre peut sauver la vie de Mister X.
* Si un détective ne peut plus avancer, les autres détectives sont toujours aux trousses de Mister X. Le détective en question deviendra un spectateur de la partie (donc il ne joue plus) jusqu’à qu’elle soit terminée.

Un déplacement est obligatoire aussi longtemps que le joueur soit en mesure de se déplacer de la situation où il se trouve i.e. en usant d’un ticket adéquat.

## Partage des responsabilités entre serveur et client

Cette section présente la répartition des responsabilités entre l’application cliente et l’application serveur. Il s’agit de décrire le rôle de chacune, leur interconnexion et leur interaction dans le déroulement du jeu.

### Client

La version cliente de l’application permet à tout un chacun de se connecter à une machine distante faisant office de serveur. Pour ce faire, un client doit spécifier les données du serveur distant d’une part i.e. le numéro de port et son adresse IP. D’autre part, il doit fournir ses informations de login i.e. nom d’utilisateur et mot de passe. La tentative de connexion sera réalisée par l’envoi d’un message au serveur. Dans l’éventualité où il ne disposerait pas de compte utilisateur, il devra en créer un en établissant une connexion avec le serveur distant. Cet établissement de connexions se fera là aussi par le biais d’un message.

L’accès à la plateforme de parties n’est possible que par la disposition d’un compte utilisateur. Cette plateforme permet à tout utilisateur de créer une partie, de rejoindre une partie, de consulter ses statistiques de jeu ou d’éditer son compte utilisateur. Elle tâchera d’afficher sous la forme d’une liste l’ensemble des parties crées ou en cours de jeu. Cette liste sera mise à jour par la réception de messages en provenance du serveur.

La création d’une partie consiste en la limitation du nombre de joueurs variant de deux à six et le choix de la carte de jeu. Les choix disponibles auront été communiqués au préalable par le serveur qui aura interrogé une base de données. Il incombe à l’utilisateur hôte de la partie de définir ces informations. La création de cette dernière n’est possible que si le serveur n’est pas plein. L’application cliente enverra donc un message au serveur lui signalant une demande de création de partie. Celui-ci répondra positivement ou négativement suivant son état. L’utilisateur peut soit tenter sa chance à nouveau soit revenir à la plateforme des parties.

Une partie crée, l’utilisateur hôte se met en attente d’utilisateurs jusqu’à ce qu’un nombre suffisant convienne. Chaque utilisateur rejoignant une partie est appelé à désigner Mister X par le biais d’un vote. Son choix est communiqué au serveur. Dès le moment où chaque utilisateur a voté, l’hôte peut lancer la partie. Un message est communiqué au serveur. Avant de faire débuter le jeu, le serveur comptabilise les votes de chaque joueur devant désigner Mister X. En cas d’égalité dans les votes, il désignera lui-même de manière aléatoire, parmi les finalistes des votes, l’utilisateur allant endosser le costume du fameux bandit. Cette information est ensuite communiquée aux utilisateurs. Le jeu peut ensuite débuter.

La partie se déroulera selon les règles du jeu établies. Chaque joueur jouera à tour de rôle. Ceux-ci seront en mesure de communiquer par le biais d’un chat durant toute la partie indépendant de leur tour de jouer. Les communications entre joueurs seront gérées par le serveur. Celui-ci veillera à transmettre les messages aux bons destinataires dans l’éventualité où certains auraient été explicitement exclus par l’émetteur.

Durant tout le déroulement du jeu, le client devra veiller au bon respect des règles du jeu. Lors de chaque message envoyé par un joueur, il sera attendu de lui qu’il contrôle que celui-ci aille dans la logique du jeu.

La main du jeu sera donnée par le serveur. Il veillera à contrôler l’état du jeu, à transmettre les informations de chaque tour à chaque joueur et de traiter les événements particuliers du jeu selon les spécificités des règles de celui.ci. En cas de fin de partie, le serveur communiquera en informera les joueurs. Il leur communiquera le camp victorieux. Ceux-ci pourront quitter la partie et revenir dans la plateforme des parties.

La possibilité qu’un joueur quitte la partie en cours de jeu devra être gérée par le serveur. Celui-ci en informera les autres joueurs. Dans le cas où il s’agirait d’un détective, la partie se poursuivra. Dans l’hypothèse où le joueur en question serait Mister X, la partie serait terminée. Le départ de Mister X serait considéré comme étant un abandon et la victoire reviendrait donc à l’équipe de détectives. Il serait également envisageable que l’hôte de la partie décide de quitter la partie. Le serveur en prendrait note et en informerait les joueurs restants. Cet événement conduirait à arrêter automatiquement la partie.

La plateforme des parties permettra non seulement de disposer d’un inventaire des parties en cours mais également de pouvoir accéder aux statistiques de jeu. Un utilisateur pourra donc disposer d’informations sur le nombre de parties qu’il aura disputé, le nombre qu’il aura remporté et le nombre qu’il aura perdu. La plateforme permettra également à tout utilisateur de pouvoir éditer son compte utilisateur. Cette édition ne portera que sur son mot de passe. Ce dernier pourra être redéfini à condition bien sûr de respecter les règles de définition de mot de passe.

La déconnexion d’un client du serveur aura pour conséquence l’envoi d’un message à celui-ci. Le serveur en prendra note et réagira de manière adéquate selon l’engagement d’un utilisateur dans une partie.

### Serveur

La version serveur de l’application permettra à tout un chacun de disposer de son propre serveur du jeu. Une personne quelconque aura donc la possibilité d’héberger un certain nombre de parties sur sa machine. Celle-ci devra néanmoins disposer de configurations matérielles suffisamment poussées de manière à pouvoir traiter une charge de travail conséquente.

Contrairement à l’accoutumée, l’application serveur disposera d’une interface graphique utilisateur. Celle-ci, destinée à un administrateur du serveur, permettra de définir le port sur lequel le serveur écoutera les requêtes des éventuels clients. Il aura également la possibilité de bloquer ou de supprimer des utilisateurs, d’ajouter et de supprimer des cartes de jeu. L’application serveur interagira étroitement avec une base de données qui lui sera locale. La machine hôte devra donc disposer en conséquence d’outils lui permettant de mettre en service une base de données MySQL. Ainsi, les utilisateurs de l’application serveur voulant mettre en service un serveur du jeu le pourront en faisant de leur propre machine une machine dédiée au serveur. Il sera donc tout à fait possible que plusieurs serveurs soient disponibles de par le monde. La connaissance de ces serveurs nécessite de connaître la machine dédiée.

Les points suivants constituent les différents rôles qu’endossera le serveur :

* Gérer les connexions clientes.

Le serveur devra être en mesure de gérer les connexions provenant de clients distants. Concrètement, il s’agira de traiter les authentifications de client avant tout accès à la plateforme de parties et à réagir de manière cohérente à toute déconnexion de client lors de n’importe quelle situation, aux conséquences qu’elle peut impliquer

* Transmettre la liste de parties hébergées sur le serveur.

La liste de parties en cours pouvant évoluer à tout moment suivant la création, la suppression ou l’arrêt d’une partie, son nouvel état devra être communiqué régulièrement aux clients distants. Il incombe au serveur de l’application d’en prendre la responsabilité.

* Gérer les parties inter clients.

Le serveur devra gérer minutieusement les différentes parties pouvant se dérouler simultanément. Il portera la casquette de gestionnaire de parties. Il devra veiller à la bonne création d’une partie, à la bonne la synchronisation des joueurs pour une partie donnée et à la bonne initialisation du jeu.

* Gérer les communications inter clients.

Durant une partie, les clients pourront s’échanger des messages. Le serveur fera office de passerelle de communication. Chaque message envoyé par un client sera transmis au serveur. Celui-ci se chargera de le retransmettre à tout le monde. Il veillera toutefois à respecter les éventuelles contraintes émises par l’émetteur. En effet, un client pouvant spécifier les destinataires du message, le serveur devra en conséquence le remettre aux utilisateurs concernés.

* Traiter les requêtes clientes destinées à la base de données.

Certaines actions exécutées par les clients nécessiteront des manipulations ou des interrogations de la base de données du jeu. Ces traitements seront réalisés par le bais de requête à destination du serveur. Ce dernier les analysera, les traitera et les transmettra à la base de données. La réponse de celle-ci sera communiquée aux clients par le serveur.

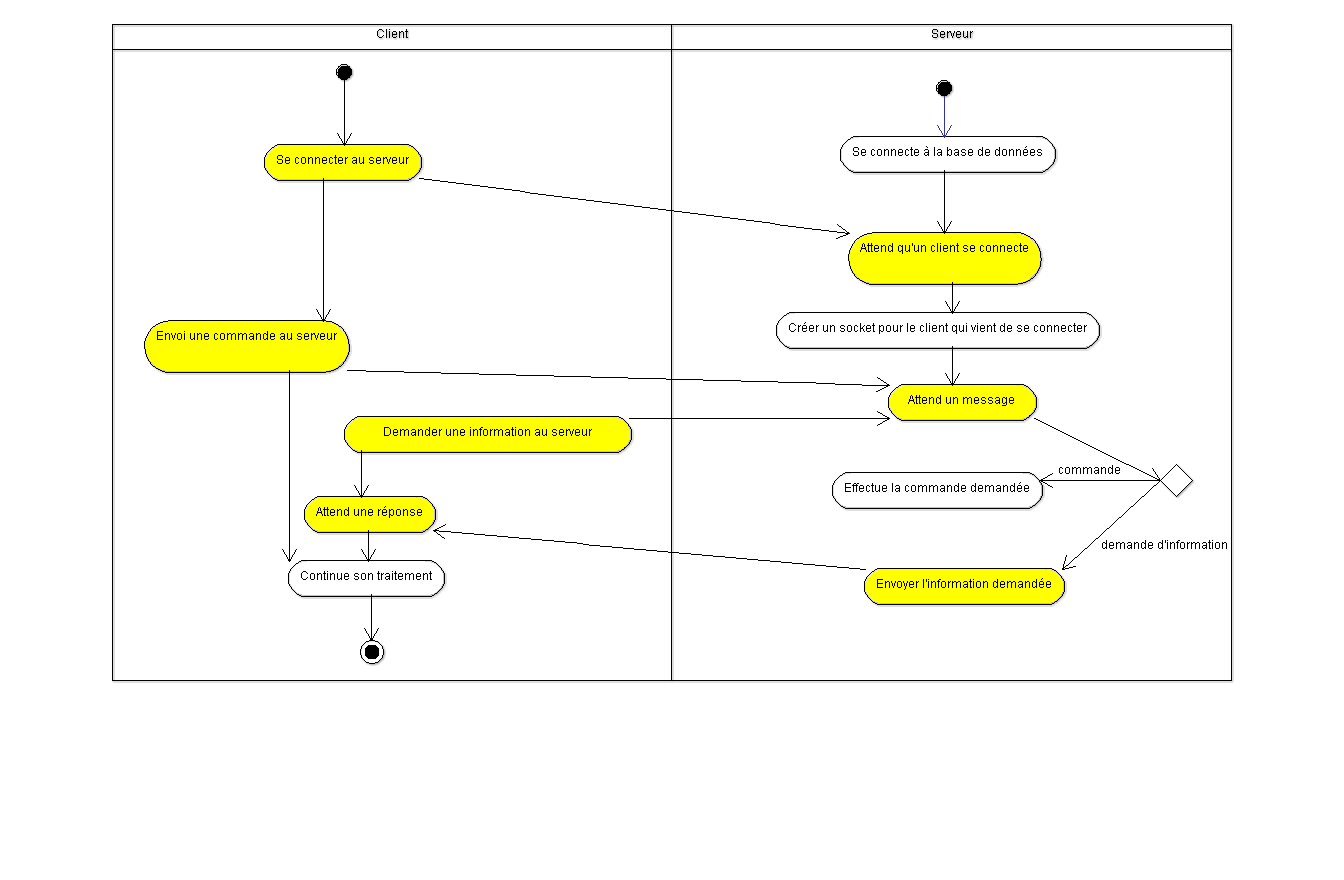
De manière générale, la tâche du serveur sera d’appliquer et de faire appliquer le protocole d’échange avec chaque client. Son rôle portera donc dans la vérification des messages envoyés par un client, par une réaction de sa part correspondant à la procédure du protocole et par l’envoi d’une réponse adéquate allant mettre à jour l’état dans lequel se trouve le client.

## Diagrammes d’activité

Nous avons décidé de faire 2 diagrammes d’activités. Un qui représente ce qui se passe entre un client et un serveur au niveau de la communication en réseau, l’autre qui représente ce qui se passe pendant une partie.

### Connexion entre client et serveur

Remarque : Les activités en jaune représente des activités asynchrones. Au vu du programme utilisé pour faire ce diagramme, il n’était pas possible d’avoir un diagramme comme présenté aux cours.



### Déroulement d’une partie

A faire…

## Cas d’utilisation

Cette section fait office de conception préliminaire du projet sous la forme de cas d’utilisation. Le diagramme général de contexte du système Scotland Yard y est exposé. Les acteurs interagissant avec le système y sont énumérés et présentés. Les scénarios des cas d’utilisations y sont détaillés.

### Diagramme général de contexte

#### Diagramme

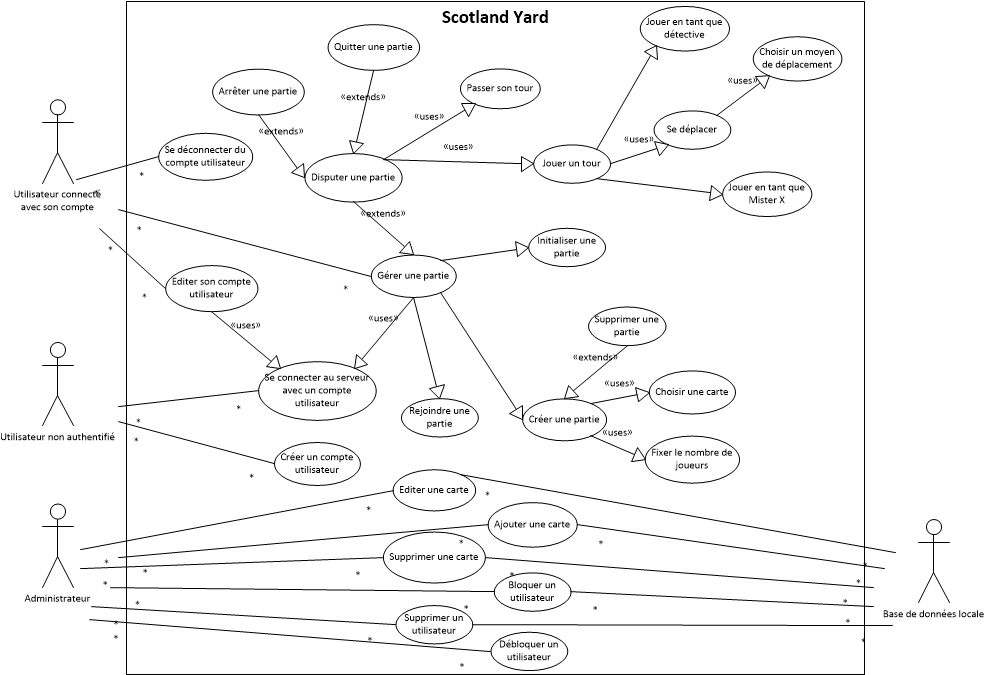
Ci-dessous figure le diagramme général de contexte du système Scotland Yard :

Figure : Diagramme général de contexte

### Description des acteurs

Ci-dessous figure la liste des acteurs du système auxquels sont associés des objectifs :

|  |  |
| --- | --- |
| **Acteur** | **Objectifs** |
| Utilisateur authentifié | * Gérer une partie. * Disputer une partie. |
| Utilisateur non authentifié | * Se connecter avec son compte utilisateur. * Créer un compte utilisateur. |
| Administrateur | * Gérer les cartes du jeu. * Gérer les utilisateurs. |
| Base de données locale | * Traiter les requêtes de l’administrateur. |

### Scénarios

#### Scénario « Se connecter au serveur avec un compte utilisateur »

Connexion au serveur avec un compte utilisateur

**Scénario principal :**

1. L’utilisateur saisit son nom d’utilisateur.
2. L’utilisateur saisit son mot de passe.
3. L’utilisateur spécifie l’adresse IP du serveur.
4. L’utilisateur spécifie le numéro de port du serveur.
5. L’utilisateur se connecte au serveur avec son compte utilisateur.

**Scénarios d’erreur :**

Opération 5 : *Si le serveur est injoignable*

1. L’utilisateur en est informé.
2. L’utilisateur vérifie les paramètres de connexion au serveur.
3. L’utilisateur tente une nouvelle connexion.

Opération 5 : *Si le login est incorrecte*

1. L’utilisateur en est informé.
2. L’utilisateur vérifie les données de son login.
3. L’utilisateur tente une nouvelle connexion.

Opération 6 : *Si l’utilisateur est bloqué*

1. L’utilisateur en est informé.

#### Scénario « Créer un compte utilisateur »

Création d’un compte utilisateur

**Scénario principal :**

1. L’utilisateur spécifie son nom d’utilisateur.
2. L’utilisateur spécifie son mot de passe.
3. L’utilisateur confirme son mot de passe.
4. L’utilisateur crée son compte.

**Scénarios d’erreur :**

Opération 1 : *Si le nom d’utilisateur ne respecte pas les règles de définition de nom d’utilisateur*

1. L’utilisateur en est informé.
2. L’utilisateur modifie son nom d’utilisateur.
3. L’utilisateur tente une nouvelle création de compte.

Opération 2 : *Si le mot de ne passe ne respecte pas les règles de définition de mot de passe*

1. L’utilisateur en est informé.
2. L’utilisateur modifie son mot de passe.
3. L’utilisateur tente une nouvelle création de compte.

Opération 4 : *Si les mots de passe ne concordent pas*

1. L’utilisateur en est informé.
2. L’utilisateur vérifie son mot de passe.
3. L’utilisateur tente une nouvelle création de compte.

Opération 5 : *Si le serveur est injoignable*

1. L’utilisateur en est informé.
2. L’utilisateur tente une nouvelle création.

Opération 5 : *Si le nom d’utilisateur est déjà utilisé*

1. L’utilisateur en est informé.
2. L’utilisateur change de nom d’utilisateur.
3. L’utilisateur tente une nouvelle création.

#### Scénario « Editer son compte utilisateur »

Edition du compte utilisateur

**Scénario principale :**

1. L’utilisateur doit être authentifié : **include Se connecter au serveur avec un compte utilisateur**
2. L’utilisateur modifie son mot de passe.
3. L’utilisateur confirme le nouveau mot de passe.
4. L’utilisateur édite son compte.
5. Le serveur met à jour la base de données.
6. Le serveur confirme les modifications.

**Scénarios d’erreur :**

Opération 1 : *Si le mot de passe ne respecte pas les règles de définition de mot de passe*

1. L’utilisateur modifie le mot de passe.
2. L’utilisateur tente d’éditer son compte.

Opération 2 : *Si les mots ne passe ne concordent pas*

1. L’utilisateur reconfirme le mot de passe.
2. L’utilisateur tente d’éditer son compte.

Opération 4 : *Si la base de données est indisponible*

1. L’utilisateur en est informé.

#### Scénario « Se déconnecter du compte utilisateur »

Déconnexion du compte utilisateur

**Scénario principale :**

1. L’utilisateur se déconnecte de son compte utilisateur.
2. Le serveur en est informé.
3. L’utilisateur est redirigé vers l’interface de connexion.

#### Scénario « Gérer une partie »

Gestion d’une partie

**Scénario principale :**

1. L’utilisateur doit être authentifié : **include Se connecter au serveur avec un compte utilisateur**
2. L’utilisateur peut :

* créer une partie : **include Créer une partie.**
* rejoindre une partie : **include Rejoindre une partie.**
* initialiser une partie : **include Initialiser une partie.**

**Scénario alternatif :**

\*1.a *La gestion d’une partie peut découler sur la dispute d’une partie.*

Un utilisateur disputer une partie : **include Disputer une partie.**

#### Scénario « Créer une partie »

Création d’une partie

**Scénario principale :**

1. L’utilisateur choisit la carte du jeu : **include Choisir une carte.**
2. L’utilisateur fixe le nombre de joueurs : **include Fixer le nombre de joueurs.**
3. L’utilisateur peut :

* supprimer une partie : **include Supprimer une partie.**
* créer la partie.

**Scénario d’erreur :**

Opération 3 : *Si le nombre de parties maximales gérées par le serveur est atteint.*

1. L’utilisateur en est informé.
2. L’utilisateur tente de créer une partie.

#### Scénario « Choisir une carte »

Choix d’une carte

**Scénario principale :**

1. L’utilisateur demande à choisir une carte.
2. Le serveur lui transmet les cartes disponibles.
3. L’utilisateur choisit une carte.

#### Scénario « Fixer le nombre de joueurs »

Fixation du nombre de joueurs

**Scénario principale :**

1. L’utilisateur fixe un nombre de joueurs.
2. L’utilisateur valide le nombre.

**Scénario d’erreur :**

Opération 1 : *Si le nombre de joueurs est insuffisant selon la règle*

1. L’utilisateur en est informé
2. L’utilisateur fixe un nouveau nombre d’utilisateurs.

Opération 1 : *Si le nombre de joueurs est trop grand selon la règle*

1. L’utilisateur en est informé
2. L’utilisateur fixe un nouveau nombre d’utilisateurs.

#### Scénario « Supprimer une partie »

Suppression d’une partie

**Scénario principale :**

1. L’utilisateur supprime la partie.
2. Le serveur en est informé.

#### Scénario « Rejoindre une partie »

Rejoint une partie

**Scénario principale :**

1. L’utilisateur sélectionne une partie.
2. L’utilisateur rejoint la partie.
3. Le serveur en est informé.
4. Le serveur informe les autres utilisateurs de la partie.

**Scénarios d’erreur :**

Opération 2 : *Si le nombre de joueurs maximal est atteint*

1. L’utilisateur en est informé.

Opération 2 : *Si la partie a déjà débuté*

1. L’utilisateur en est informé.

#### Scénario « Lancer une partie »

Initialisation d’une partie

**Scénario principal :**

1. L’utilisateur hôte attend des utilisateurs.
2. L’utilisateur hôte lance la partie.

**Scénario d’erreur :**

Opération 2 : *Si le nombre de joueurs est insuffisant*

1. L’utilisateur hôte en est informé.
2. L’utilisateur hôte attend des joueurs supplémentaires.
3. L’utilisateur hôte lance la partie.

#### Scénario « Disputer une partie »

Dispute d’une partie

**Scénario principale :**

1. L’utilisateur reçoit les données d’initialisation du jeu :
   1. La carte.
   2. Les stations.
   3. Les liens entre stations
   4. Les tickets de chaque joueur.
   5. La position de chaque joueur sauf celle de Mister X
2. L’utilisateur joue son tour : **include Jouer un tour**.

**Scénario alternatif :**

\*2.a *A tout moment, durant la dispute de la partie*

Un utilisateur tchate : **include Tchater.**

\*2.c *A tout moment, durant la dispute de la partie*

Un utilisateur quitte la partie : **include Quitter une partie.**

\*2.c *A tout moment, durant la dispute de la partie*

L’utilisateur hôte arrête la partie : **include Arrêter une partie.**

#### Scénario « Quitter une partie »

Sortie de la partie

**Scénario principale :**

1. Le détective quitte la partie.
2. Le serveur en est informé.
3. Le serveur informe les autres joueurs.

**Scénario alternatif :**

1. Mister X quitte la partie.
2. Le serveur en est informé.
3. Le serveur informe les détectives.
4. La partie est terminée.

#### Scénario « Arrêter une partie »

Arrêt de la partie

**Scénario principale :**

1. Le joueur hôte arrête la partie.
2. Le serveur en est informé.
3. Le serveur informe les autres joueurs.
4. La partie est arrêtée.

#### Scénario « Passer son tour »

Passé son tour de jouer.

**Scénario principal :**

1. Le joueur passe son tour.
2. Le serveur en est informé.
3. Le serveur informe tous les joueurs.

**Scénarios d’erreur :**

Opération 1 : *Si le joueur n’a pas la main*

1. Le joueur en est informé.
2. Le joueur attend son tour.

#### Scénario « Jouer un tour »

Tour de jeu d’un joueur

Scénario principal :

1. Le joueur se déplace : **include Se Déplacer**
2. Le joueur joue son tour :

* En tant que détective : **include Jouer un tour en tant que détective.**
* En tant que Mister X : **include Jouer un tour en tant que Mister X.**

**Scénarios d’erreur :**

Opération 1 : *Si le joueur n’a pas la main*

1. Le joueur en est informé.
2. Le joueur attend son tour.

Opération 1 : *Si le joueur ne peut plus se déplacer*

1. Le joueur en est informé.
2. Le joueur passe son tour.

#### Scénario « Jouer un tour en tant que détective »

Tour de jeu d’un détective

**Scénario principal :**

1. Le détective joue son tour.
2. Le détective met à jour ses tickets.
3. Le serveur en est informé.
4. Le serveur communique le jeu du détective aux autres joueurs.
5. Mister X reçoit le ticket dépensé par le détective.

#### Scénario « Jouer un tour en tant que Mister X »

Tour de jeu de Mister X

**Scénario principal :**

1. Mister X se déplace : **include Se déplacer**
2. Mister X joue son tour.
3. Mister X met à jour ses tickets.
4. Le serveur en est informé.
5. Le serveur communique le moyen de déplacement de Mister X aux détectives.
6. Le ticket utilisé est perdu.

**Extensions :**

5a. Apparition de Mister X

Le serveur communique la position de Mister X aux détectives aux tours 3, 8, 13 et 18.

5b. Pas de communication

Mister X a joué le ticket demandant de cachant son moyen de déplacement.

5c. Double tour

Mister X a joué le ticket lui permettant de jouer un tour consécutif.

#### Scénario « Se déplacer »

Déplacement dans la carte

**Scénario principale :**

1. Le joueur choisit un moyen de déplacement : **include Choisir un moyen de déplacement.**
2. Le joueur choisit une station.
3. Le joueur confirme ses choix.
4. Le serveur en est informé.

**Scénario d’erreur :**

Opération 2 : *Si la station n’est pas accessible*

1. Le joueur en est informé.
2. Le joueur choisit une autre station.

#### Scénario « Choisir un moyen de déplacement »

Choix d’un moyen de déplacement

**Scénario principale :**

1. Le joueur choisit un moyen de déplacement.

Opération 1 : *Si le ticket relatif au moyen de déplacement est épuisé*

1. Le joueur en est informé.
2. Le joueur choisit un autre moyen de déplacement.

#### Scénario « Bloquer un utilisateur »

Blocage d’un utilisateur

**Scénario principale :**

1. L’administrateur sélectionne un utilisateur.
2. L’administrateur bloque l’utilisateur sélectionné.

**Scénario d’erreur :**

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.

#### Scénario « Débloquer un utilisateur »

Blocage d’un utilisateur

**Scénario principale :**

1. L’administrateur sélectionne un utilisateur.
2. L’administrateur débloque l’utilisateur sélectionné.

**Scénario d’erreur :**

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.

#### Scénario « Supprimer un utilisateur »

Suppression d’un utilisateur

**Scénario principale :**

1. L’administrateur sélectionne un utilisateur.
2. L’administrateur supprime l’utilisateur sélectionné.

**Scénario d’erreur :**

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.

Opération 2 : *Si l’utilisateur est connecté.*

1. L’administrateur en est informé.

#### Scénario « Ajouter une carte »

Edition d’une carte

**Scénario principale :**

1. L’administrateur crée une carte.

**Scénario d’erreur :**

Opération 2 : *Si les règles de création de carte ne sont pas respectées*

1. L’administrateur en est informé.

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.

#### Scénario « Supprimer une carte »

Suppression d’un utilisateur

**Scénario principale :**

1. L’administrateur sélectionne une carte.
2. L’administrateur supprime la carte sélectionnée.

**Scénario d’erreur :**

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.

#### Scénario « Editer une carte »

Edition d’une carte

**Scénario principale :**

1. L’administrateur sélectionne une carte.
2. L’administrateur édite la carte sélectionnée.

**Scénario d’erreur :**

Opération 2 : *Si les règles d’édition de carte ne sont pas respectées*

1. L’administrateur en est informé.

Opération 2 : *Si la base de données est indisponible*

L’administrateur en est informé.

## Base de données

Cette section introduit la base de données allant être utilisées par l’application. Celle-ci interagira directement avec le serveur.

L’application cliente désirant interroger la base de données devra passer par le serveur. Celui-ci se verra confier la tâche de transmettre la requête du client et la réponse de la base de données.

Les réponses de la base de données que tâchera de transmettre le serveur au client seront envoyées par sérialisation.

### Objectifs

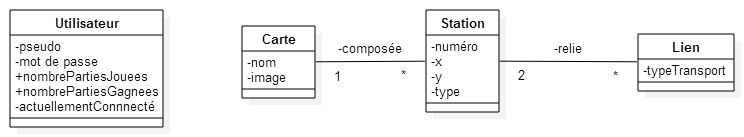
Les points suivants constituent les objectifs que la base de données devra remplir en termes de stockage de données persistantes :

* Comptes utilisateurs
  + Les informations relatives à un compte utilisateur comprennent nom d’utilisateur et mot de passe.
* Cartes du jeu

Les informations relatives aux cartes du jeu comprennent une image constituant l’arrière-plan de la carte et un ensemble de stations.

### Modèle entité-association (VANESSA)

Ci-dessous figure le modèle entité-association de la base de données :



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LE MODELE EST FAUX, IL FAUT RETIRER LE NOMBRE DE PARTIES JOUEES ET LE NOMBRE DE PARTIES GAGNEES, AJOUTER UN ATTRIBUT BOOLEEN BLOCKED DANS L’ENTITE UTILISATEUR. IL FAUT EGALEMENT TOUT RENOMMER EN ANGLAIS. POUR FAIRE COURT, IL FAUT SE REFERER AU SCRIPT SQL.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Contraintes d’intégrité

Les points suivants constituent les contraintes d’intégrité du modèle :

* Le pseudo d'un utilisateur ne doit pas contenir de # (pour ne pas corrompre le protocole)
* Le mot de passe d'un utilisateur ne doit pas contenir de # (même raison)
* Le nom d'une carte ne doit pas contenir de # (même raison)
* Un lien doit relier 2 stations différentes, et en plus ces deux stations doivent se trouver sur la même carte.
* Si un lien existe entre une station A et une station B, son type de transport doit être différents des autres liens reliant A et B
* Il ne peut pas y avoir deux stations avec le même numéro dans une carte
* Les coordonnées x et y d'une station sont positives ou nulles
* Le type d'une station peut prendre uniquement les valeurs : TAXI TAXIBUS TAXIBUSMETRO
* Le type de transport d'un lien peut prendre uniquement les valeurs : TAXI BUS METRO
* Un lien de type BUS peut relier uniquement deux stations ayant comme type : TAXIBUS ou TAXIBUSMETRO
* Un lien de type METRO peut relier uniquement deux stations ayant comme type : TAXIBUSMETRO

Il est à noter que la grande majorité des contraintes énumérées sera gérée dans l’implémentation de la solution plutôt que du côté de la base de données.

### Commentaires

Les points suivants constituent quelques explications à titre indicatives relatives au modèle entité-association de la base de données :

* Un utilisateur est caractérisé par un pseudo et un mot de passe. Il dispose également de deux attributs booléens indiquant respectivement s’il est actuellement connecté et si son compte est couramment bloqué.
* Une carte est caractérisée avant tout par un nom et par une image. Celle-ci constitue l’arrière-plan de la carte. Une carte est également composée par un ensemble de stations. Ces dernières sont caractérisées par un numéro et des coordonnées x et y. Une station est définie par un type : StationTaxi, StationBusTaxi ou StationMetroiBusTaxi.

## Modèle de domaine

Cette partie du document présente les des modèles de domaine devant représenter l’application. Celle-ci se divise en deux modèles de domaine distincts. Le premier a pour but de représenter le serveur. Le second a pour vocation d’illustrer le client.

### Modèle de domaine côté serveur

#### Modèle

Ci-dessous figure une ébauche du modèle de domaine côté serveur pour lequel aucune contrainte d’intégrité n’a été définie temporairement :

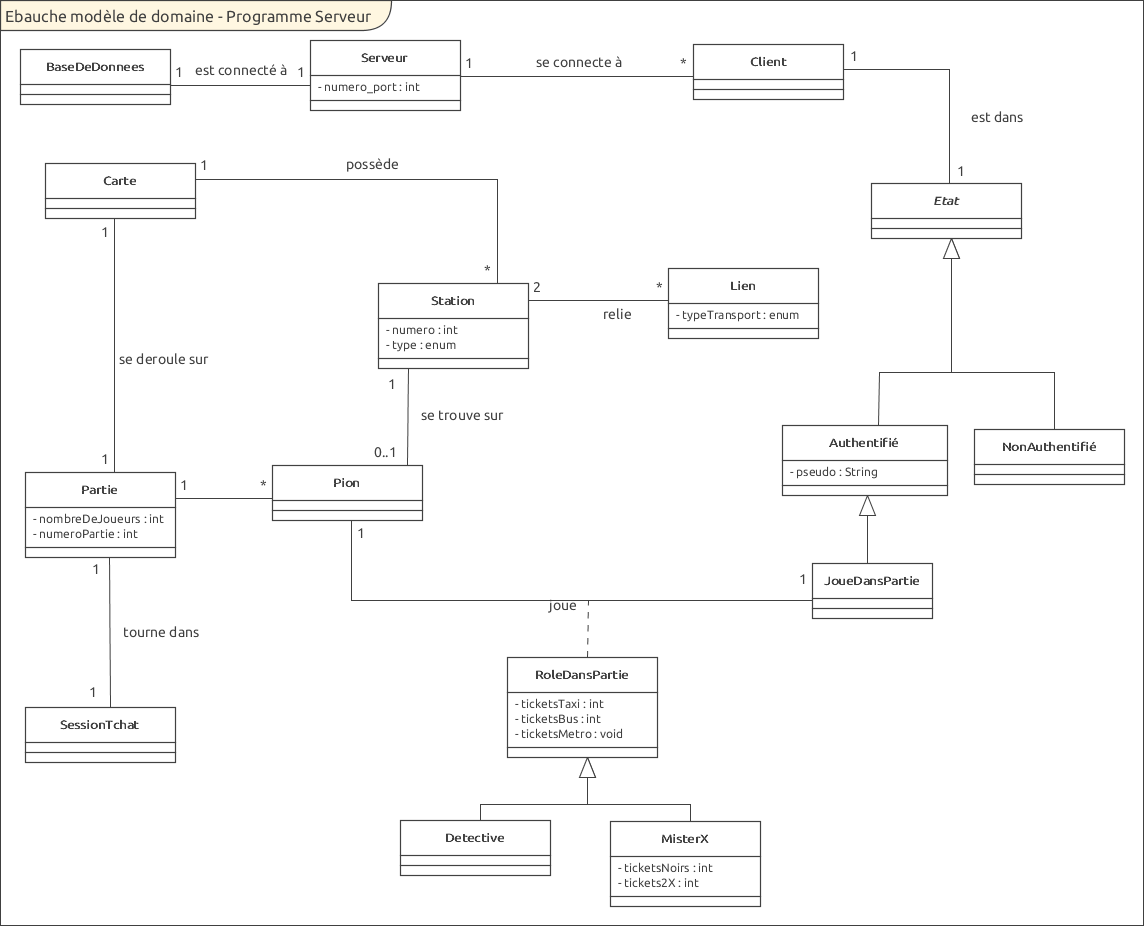


Figure : Modèle de domaine côté serveur

#### Commentaires

Le serveur est connecté à une base de données. C’est elle qui lui fournira les informations sur des cartes ou bien des comptes utilisateurs.

Plusieurs clients peuvent se connecter à un serveur. Chaque client connecté aura un état vis-à-vis du serveur, qui peut être l’un des états suivants :

* Non authentifié
  + Signifie que le client n’est pas authentifié sur un compte utilisateur
* Authentifié
  + Signifie que le client est authentifié sur un compte utilisateur
* En train de jouer dans une partie
  + Signifie que le client est authentifié est qu’il est en train de jouer dans une partie

Lorsqu’un client se trouve dans une partie, il est rattaché à un pion qui le caractérise comme jouant soit le rôle de MisterX, soit le rôle d’un détective. Que ça soit un détective ou Mister X, il possède un certain nombre de tickets de taxi, bus, métros pendant la partie.

Lors du déroulement de la partie, un pion se trouve sur l’une des stations se trouvant sur la carte. Une partie se déroule sur une seule carte.

Entre deux stations A et B, il peut y avoir un lien qui symbolise le fait qu’il est possible d’aller de A à B ou de A à B. Ce lien est caractérisé par un type de moyen de transport (par exemple Bus qui signifierait que ce lien permet d’aller de A à B en bus…).

Finalement, il y a une session de tchat qui est lancée dans chaque partie, afin que les joueurs puissent discuter. Chaque partie possède sa propre session de tchat.

### Modèle de domaine côté client

#### Modèle

Ci-dessous figure une ébauche du modèle de domaine côté client pour lequel aucune contrainte d’intégrité n’a été définie temporairement :

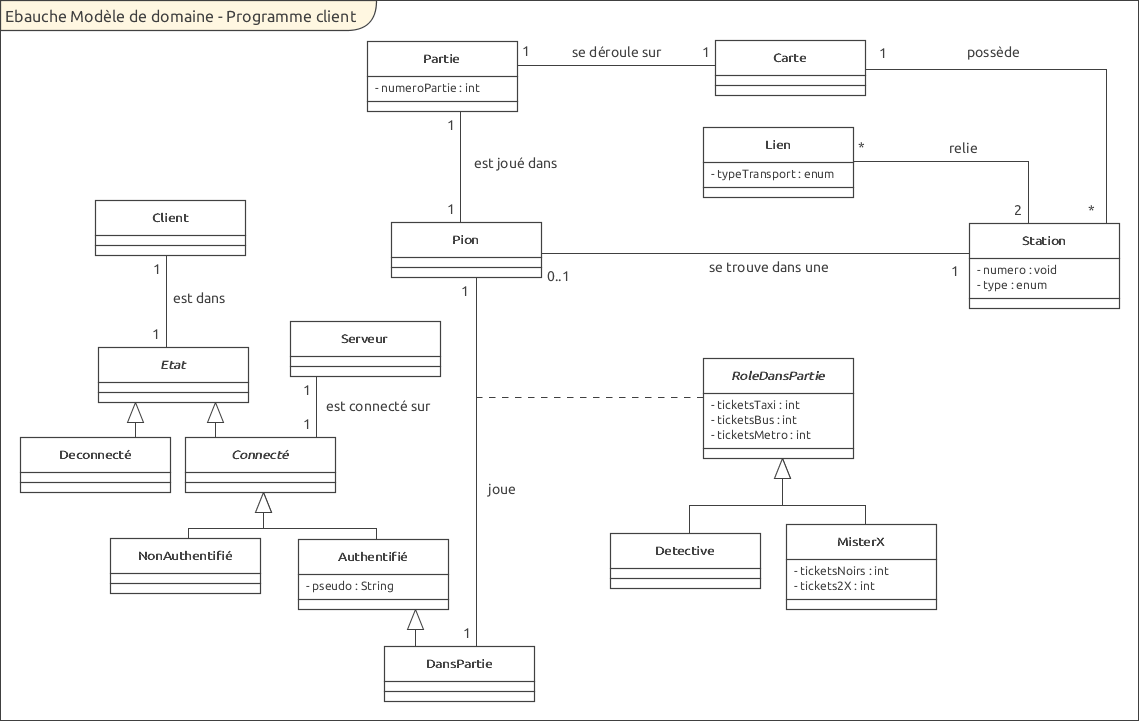


Figure : Modèle de domaine côté client

#### Commentaires

Un client a plusieurs états possibles qu’il peut prendre à la fois. Ces différents états sont les suivants :

* Déconnecté
  + Signifie que le client n’est pas du tout connecté à un serveur.
* Connecté
  + Signifie que le client est connecté à un serveur.
* Connecté et non authentifié
  + Signifie que le client est connecté mais qu’il n’est pas authentifié sur son compte utilisateur.
* Connecté et authentifié
  + Signifie que le client est connecté et qu’il est authentifié sur son compte utilisateur.
* Dans une partie
  + Signifie que le client est connecté, authentifié et qu’il est en train de jouer une partie.

Concernant les autres entités, voir les explications du modèle de domaine du serveur.

# Conception

## Protocole d’échange entre le client et le serveur (RAPHAEL)

Voici le protocole d’échange qui sera utilisé pour la communication entre le client et le serveur.

La syntaxe de base est : **COMMANDE#**param1**#**param2**#**...**#**paramN

| **Commande** | **Direction** | **Description** |
| --- | --- | --- |
| PLAY\_DETECTIVE\_TURN  #pseudo  #numeroStation  #transport  #numeroPartie | Client 🡪 Serveur | Commande qui permet au client de dire qu’il effectue un déplacement (donc joue un coup) d’un détective dans la partie ayant le numéro **#numeroPartie**.  Dans ce message se trouve également le numéro de la station où le détective s’est déplacé, le transport qu’il a utilisé (afin que le serveur puisse comptabiliser les tickets) ainsi que le pseudo du joueur qui a joué.  **Exemple :** PLAY\_DETECTIVE\_TURN#toto123#156#metro#2  *« Le détective qui a le pseudo toto123 s’est déplacer sur la station 156 en prenant le métro, sur la partie possédant l’identifiant 2 »* |
| PLAY\_MISTER\_X\_TURN  #pseudo  #numeroStation  #transport  #ticketDouble  #numeroPartie | Client 🡪 Serveur | Commande qui permet au client de dire qu’il effectue un déplacement (donc joue un coup) de Mister X dans la partie ayant le numéro **#numeroPartie**.  Dans ce message se trouve également le numéro de la station où Mister X s’est déplacé, le transport qu’il a utilisé (afin que le serveur puisse comptabiliser les tickets) ainsi que le pseudo du joueur qui a joué.  Le message contient aussi une information qui dit si Mister X a utilisé un ticket pour jouer 2 fois ou bien pas.  **Exemple :** PLAY\_MISTER\_X\_TURN#boulette#32#noir#true#3  *« Mister X qui a le pseudo boulette s’est déplacé sur la station 32 en prenant un ticket noir (donc n’importe quel moyen de transport) signifiant qu’il veut cacher son moyen de transport, sur la partie possédant l’identifiant 3 »* |
| NOTIFY\_DETECTIVE\_PLAYED  #pseudo  #prochainJoueurPseudo  #ancienJoueurPseudo  #transport  #station  #numeroPartie | Serveur 🡪 Client | Commande qui permet au serveur de dire au client qu’un joueur détective s’est déplacé dans la partie qui a l’identifiant #numeroPartie.  Dans ce message, on y trouve le pseudo a qui le message est envoyé (comme ça le client peut vérifier que c’est bien pour lui), moyen de transport que le détective précédant a utilisé, le pseudo de celui qui vient de jouer, la station sur laquelle s’est rendue le détective, le pseudo du prochain joueur qui doit jouer (comme ça le client peut savoir s’il doit jouer ou non).  **Exemple :**  NOTIFY\_DETECTIVE\_PLAYED#bouboule#bouboule#yakahoma#taxi#127#3  *« Cette notification est destinée à bouboule. Bouboule doit maintenant jouer son tour et le joueur yakahoma vient de se déplacer en taxi vers la station 127 dans la partie ayant l’identifiant 3 »* |
| NOTIFY\_MISTER\_X\_PLAYED  #pseudo  #prochainJoueurPseudo  #ancienJoueurPseudo  #transport  #station  #numeroPartie | Serveur 🡪 Client | Commande qui permet au serveur de dire au client que Mister X a joué son coup et s’est déplacé dans la partie qui a l’identifiant #numeroPartie.  Dans ce message, on y trouve le pseudo a qui le message est envoyé (comme ça le client peut vérifier que c’est bien pour lui), moyen de transport que le détective précédant a utilisé, le pseudo de celui qui vient de jouer, la station sur laquelle s’est rendue le détective, le pseudo du prochain joueur qui doit jouer (comme ça le client peut savoir s’il doit jouer ou non).  A noter que le client doit afficher l’emplacement de Mister X si le paramètre station est différent de -1. Dans le cas échéant, la position de Mister X ne doit pas être affichée.  **Exemple :**  NOTIFY\_MISTER\_X\_PLAYED#mickey#donald#georges#noir#-1#5  *« Cette notification est destinée à Mickey. Donald doit maintenant jouer son tour et le joueur Georges vient de se déplacer (en cachant son moyen de transport car il utilise un ticket noir) vers une station inconnu (dû au -1), dans la partie ayant l’identifiant 3 »* |
| LAUNCH\_GAME  #pseudo  #nombreJoueurs  #carte | Client 🡪 Serveur | Commande qui permet à un client de lancer une partie sur le serveur.  Elle contient le nombre de joueurs de la partie ainsi que la carte qui auront été choisi par celui qui lance la partie, ainsi que le pseudo de l’utilisateur qui veut lancer une partie.  **Exemple :**  LAUNCH\_GAME#aladdin#5#londres  *« Demande de lancement d’une nouvelle partie de la part de Aladdin avec 5 joueurs et pour jouer sur la carte de Londres »* |
| ACCEPT\_CREATION\_GAME  #pseudo  #numeroPartie | Serveur 🡪 Client | Suite à une demande de lancement d’une partie (commande LAUNCH\_GAME) le serveur renverra ce message à l’utilisateur ayant le pseudo #pseudo et le numéro de la partie qui a été créer sur le serveur, afin que le client connaisse le numéro de cette partie.  **Exemple :**  ACCEPT\_CREATION\_GAME#bouboule#4  *« Notification à destination de bouboule. Le serveur a accepté la demande d’une nouvelle partie bouboule et cette partie a le numéro 4 »* |
| REJECT\_CREATION\_GAME  #pseudo | Serveur 🡪 Client | Permet simplement de dire au client ayant le pseudo #pseudo que la création de la partie est refusée.  **Exemple :**  REJECT\_CREATION\_GAME#tintin  *« Notification pour Tintin. La demande de création d’une partie a été refusée »* |
| IMAGESIZE  #taille | Serveur 🡪 Client | Permet au serveur de transmettre la taille d’une image à un client afin qu’il connaisse la taille de l’image à recevoir. |
| STARTGAME  #nompartie | Client 🡪 Serveur | Permet à l’hôte de la partie de dire au serveur de démarrer la partie car elle est prête au niveau des joueurs. |
| AUTHENTICATE  #username  #password | Client 🡪 Serveur | Permet à un client de s’authentifier sur le serveur avec un **username** et un **password** |
| UNAUTHENTICATE | Client 🡪 Serveur | Permet à un client de de déconnecter de son compte sur le serveur. |
| CREATEACCOUNT  #username  #password | Client 🡪 Serveur | Permet à un client de créer un compte utilisateur sur le serveur avec un **username** et un **password**. |
| REQUESTMAPLIST | Client 🡪 Serveur | Permet à un client de demander au serveur la liste des noms des cartes disponibles |
| REQUESTGAMELIST | Client 🡪 Serveur | Permet à un client de demander au serveur la liste des parties en cours |
| REQUESTPLAYERLIST  #nompartie | Client 🡪 Serveur | Permet à un client de demander la liste des joueurs d’une partie (portant le nom **nompartie**) |
| EDITACCOUNT  #newUsername  #newPassword | Client 🡪 Serveur | Permet à un client d’éditer les informations de son compte soit sont username et son password. |

Pour le reste des informations utiles à transmettre (cartes, stations, etc…) ceci se fait par sérialisation JSon (voir le chapitre concerné).

## Serveur

Voici la partie conception en ce qui concerne le serveur.

### Architecture

Pour gérer l’état d’un client vis-à-vis du serveur ainsi que l’état d’une partie, nous avons décidé d’utiliser le modèle de conception réutilisable State Pattern, qui nous a facilité pas mal de traitements.

Le design pattern State consiste à faire abstraction du comportement qu’un objet doit avoir suivant dans quel état il est (par exemple si on a une Voiture qui est en train de rouler, elle ne fera pas la même chose que si elle est à l’arrêt), la classe Voiture dans ce cas ferait abstraction de son état car elle déléguerait ses traitements à son état, qui gère lui-même ce que la voiture doit faire…

Au niveau des parties, nous avons 3 états :

* Création de la partie (attente des joueurs)
* Initialisation de la partie (préparation des tickets, envoi de la carte aux joueurs, etc.)
* Déroulement de la partie

Au niveau des clients nous avons 3 états :

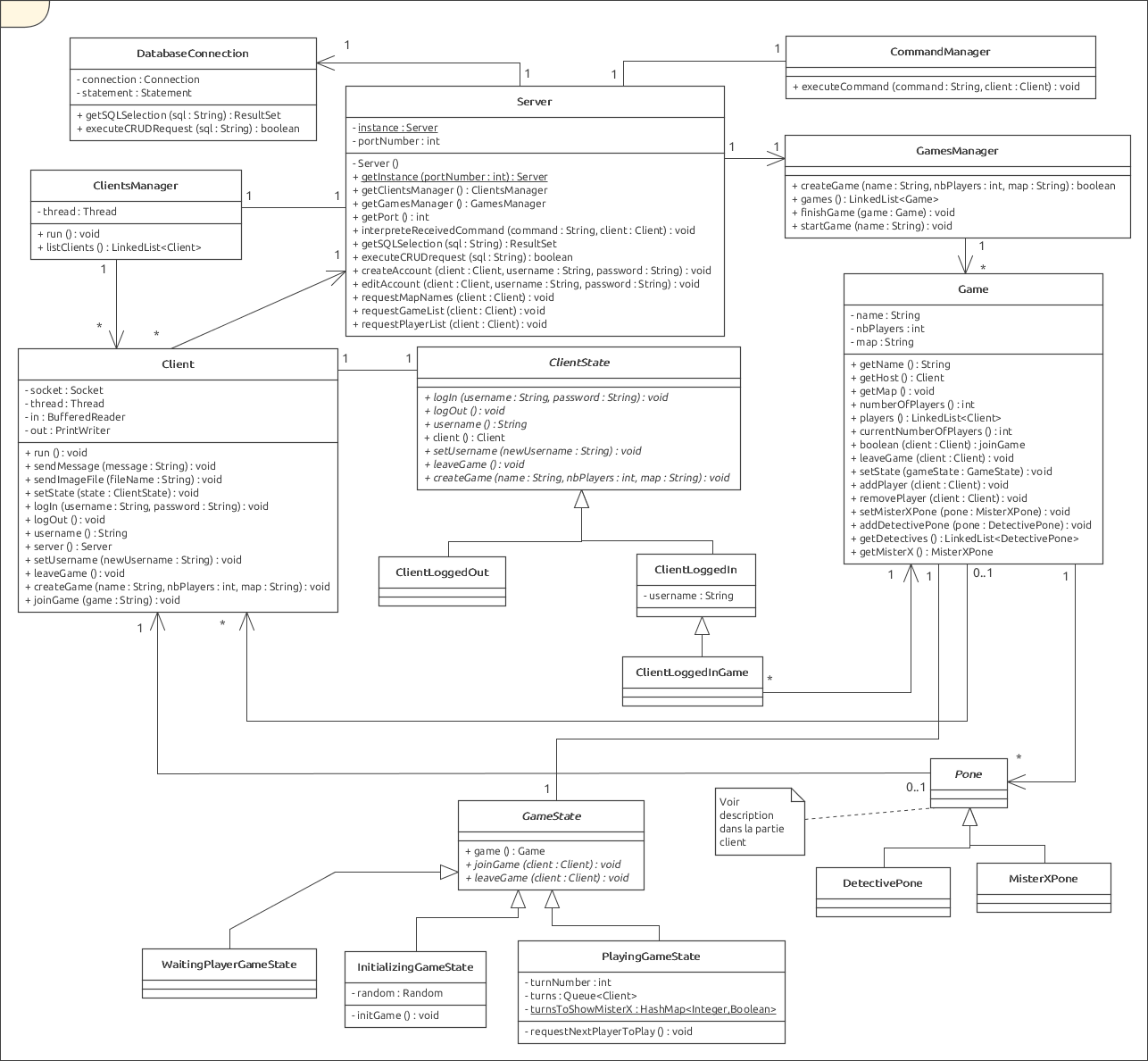
* Le client est connecté mais n’est pas authentifié
* Le client est connecté et authentifié
* Le client est connecté, authentifié et dans une partie

Au niveau du serveur, nous avons 3 objets principaux :

* Un Client Manager qui attend que les clients se connectent
* Un Command Manager qui traite les commandes reçues par les clients
* Un Game Manager qui gère la liste des parties lancées sur le serveur

### Diagramme de classes

Voici le diagramme de classes du serveur. Pour des raisons de simplification, les constructeurs ont été retirés, et les classes concernant les formulaires graphiques ne sont pas montrées.



Concernant la description des classes, consulter les fichiers sources (commentaires).

### Packages

* **scotlandyardserver** 🡪 Contient les fichiers principaux du projet
  + Server.java
  + Serveur\_Scotland\_Yard.java
  + Utils.java
* **scotlandyardserver.admin.map** 🡪 Contient les classes permettant de créer, éditer une carte
  + GUIAddMap.java
  + GUIAddStation.java
  + GUIAddStationLink.java
  + GUIEditMap.java
  + GUIMap.java
* **Scotlandyardserver.admin.user** 🡪 Contient les classes permettant de créer, éditer un utilisateur
  + GUIAddUser.java
  + GUIEditUser.java
  + GUIUser.java
* **Scotlandyardserver.client** 🡪 Contient les classes permettant de gérer les clients qui se connectent au serveur
  + Client.java
  + ClientsManager.java
* **Scotlandyardserver.client.state** 🡪 Contient les classes permettant de gérer l’état d’un client
  + ClientLoggedIn.java
  + ClientLoggedInGame.java
  + ClientLoggedOut.java
  + ClientState.java
* **Scotlandyardserver.commands** 🡪 Contient la classe permettant de traiter la réception des commandes depuis un client
  + CommandManager.java
* **Scotlandyardserver.database** 🡪 Contient la classe permettant la connexion à la base de données
  + DatabaseConnection.java
* **Scotlandyardserver.games** 🡪 Contient les classes qui concernent une partie
  + DetectivePone.java
  + Game.java
  + GamesManager.java
  + MisterXPone.java
  + Pone.java
* **Scotlandyardserver.games.state** 🡪 Contient les classes qui concernent la gestion de l’état d’une partie
  + GameState.java
  + InitializingGameState.java
  + PlayingGameState.java
  + WaitingPlayerState.java
* **Scotlandyardserver.json** 🡪 Contient les classes permettant la sérialisation JSon
  + Game.java
  + GameData.java
  + GameList.java
  + GameMap.java
  + Link.java
  + MapNames.java
  + MisterXData.java
  + PlayerData.java
  + PlayerDataList.java
  + PlayerList.java
  + Station.java

## Client (YASSIN, VANESSA)

### Architecture(YASSIN)

PARLER ENTRE AUTRE DU STATE PATTERN

### Diagramme de classes(YASSIN OU VANESSA)

### Description(YASSIN, VANESSA)

IL FAUT SE PARTAGER LES PACKAGES ET LES CLASSES, S’INFORMER REGULIEREMENT SUR QUELLES CLASSES CHACUN TRAVAIL

### Package XYZ

#### Classe XYZ

##### Interface

##### Description

## Base de données (VANESSA)

### Modèle relationnel

Ci-dessous figure le modèle relationnel de la base de données :

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CRÉER LE MODELE RELATIONNEL DE LA BASE DE DONNEES, C’EST-A-DIRE, QU’ON PUISSE VOIR LE DETAIL : CLES, TYPES, AUTRES…

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### Script

Ci-dessous figure le script SQL permettant la création de la base de données du jeu :

|  |  |
| --- | --- |
| **1**  **2**  **3**  **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20**  **21**  **22**  **23**  **24**  **25**  **26**  **27**  **28**  **29**  **30**  **31**  **32**  **33**  **34**  **35**  **36**  **37**  **38**  **39**  **40**  **41**  **42**  **43** | **SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;**  **SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;**  **SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL';**  **DROP SCHEMA IF EXISTS scotlandyard;**  **CREATE SCHEMA scotlandyard;**  **USE scotlandyard;**  **CREATE TABLE user (**  **username VARCHAR(20) NOT NULL,**  **password VARCHAR(20) NOT NULL,**  **connected BOOLEAN DEFAULT FALSE,**  **blocked BOOLEAN DEFAULT FALSE,**  **PRIMARY KEY (username)**  **)ENGINE=InnoDB DEFAULT CHARSET=utf8;**  **CREATE TABLE map (**  **name VARCHAR(20) NOT NULL,**  **picture VARCHAR(100) NOT NULL,**  **PRIMARY KEY (name)**  **)ENGINE=InnoDB DEFAULT CHARSET=utf8;**  **CREATE TABLE station (**  **station\_id INT UNSIGNED NOT NULL,**  **number INT UNSIGNED NOT NULL,**  **x INT UNSIGNED NOT NULL,**  **y INT UNSIGNED NOT NULL,**  **type ENUM('taxi', 'taxibus', 'taxibussubway'),**  **map\_fk VARCHAR(20) NOT NULL,**  **PRIMARY KEY (station\_id),**  **CONSTRAINT `fk\_station\_map` FOREIGN KEY (map\_fk) REFERENCES map (name) ON DELETE CASCADE ON UPDATE CASCADE**  **)ENGINE=InnoDB DEFAULT CHARSET=utf8;**  **CREATE TABLE link (**  **first\_station\_fk INT UNSIGNED NOT NULL,**  **second\_station\_fk INT UNSIGNED NOT NULL,**  **type ENUM('taxi', 'bus', 'subway') NOT NULL,**  **PRIMARY KEY(first\_station\_fk, second\_station\_fk, type),**  **CONSTRAINT `fk\_link\_first\_station` FOREIGN KEY (first\_station\_fk) REFERENCES station (station\_id),**  **CONSTRAINT `fk\_link\_second\_station` FOREIGN KEY (second\_station\_fk) REFERENCES station (station\_id)**  **)ENGINE=InnoDB DEFAULT CHARSET=utf8;** |

## Structures Json (YASSIN)

### XXX

#### Structure

#### Description

## Règles de définition de compte utilisateur

Cette partie présente les règles de définition de compte utilisateur. Il s’agit de présenter en détails les règles qui régissent la définition d’un nom d’utilisateur et d’un mot de passe utilisateur.

### Définition de nom d’utilisateur

Un nom d’utilisateur doit respecter les contraintes suivantes :

* Doit contenir entre six et vingt caractères.
* Doit contenir aucun caractère #.
* Doit contenir aucun espace.
* Peut contenir des caractères numériques.
* Peut contenir des caractères alphabétiques.
* Peut contenir des caractères spéciaux.

L’expression régulière suivante veille à faire respecter ces contraintes :

* ^(?=.\*.)(?!.\*[#|\s]).{6,20}$

### Définition de mot de passe

Un mot de passe utilisateur doit respecter les contraintes suivantes :

* Doit contenir entre six et vingt caractères.
* Doit contenir aucun caractère #.
* Doit contenir aucun espace.
* Doit contenir au moins un caractère numérique.
* Doit contenir au moins un caractère alphabétique minuscule.
* Doit contenir au moins un caractère alphabétique majuscule.
* Doit contenir au moins un caractère spécial.

L’expression régulière suivante veille à faire respecter ces contraintes :

* ^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])(?!.\*[#|\s]).{6,20}$

# Implémentation

## Technologies (Vanessa)

## Client(VANESSA)

Présenter les interfaces graphiques et les décrire

## Serveur(Yassin)

Présenter les interfaces graphiques et les décrire

## Problèmes rencontrés (Raphaël)

Malheureusement, un problème dans le code nous a empêchés de finir l’application à temps.

En effet, nous avons découvert un bug de dernière minute, qui nous a sauvagement posé problème pour le peu de choses qu’il restait à développer. Nous avons constaté que du côté client la réception des messages et mal gérée (le client ne reçoit des fois rien du tout alors que le serveur a envoyé des informations), ceci nous empêche donc de gérer le déroulement du jeu correctement.

Comme ce bug est très difficile à corriger et n’a pas été découvert tout de suite (il s’agit sans doute d’un problème de modélisation de ce côté-là, nous n’avons pas pu développer le déroulement de la partie en soit).

Il est à noter que si ce bug n’était pas présent, on aurait su ce qu’il restait à développer pour finir le programme.

C’est le seul problème que nous avons rencontrés dans le projet tout au long des itérations, comme quoi les bugs de dernière minutes (parfois grave ce qui a été le cas dans notre projet) peuvent conduire à un échec qui est très difficile à combler au vu du peu de temps qu’il restait pour finir le tout.

# Gestion du projet

## Rôle des participants

Il a été décidé que le projet se déroulerait sans chef de projet de manière ce que chacun n’ait de compte à rendre à qui que ce soit au sein du groupe.

### Guillaume Bruge

### Yassin Kammoun (YASSIN)

### Michelle Meguep (VANESSA)

### Raphäel Racine (RAPHAEL)

ECRIRE QUOI ?

## Plan d’itérations initial

Ci-dessous est décrit le plan d’itérations initial défini au début du projet. Celui-ci se décompose en huit itérations distinctes. Chacune a pour objectif de développer un ensemble de fonctionnalités correspondant à des scénarios, chaque scénario appartenant à un certain cas d’utilisation.

### Itération 1 : 22.04.2015 - 29.04.2015

#### Objectif

La première itération consistera en la mise en place de l’architecture de base du programme client-serveur et de la base de données du jeu.

#### Description

Les points suivants constituent le descriptif détaillé du contenu du travail effectué lors de la première itération:

* Base de données

La base de données du jeu devra être créée conformément en modèle conceptuel établi. Cela consistera à définir les tables, les procédures stockées et les éventuelles contraintes d’intégrité. Celle-ci devra également faire l’objet d’un dumping de données. Celui-ci portera sur la création de la carte par défaut du jeu.

* Protocole client-serveur

Le protocole client-serveur devra être implémenté. Cela consistera à concevoir la totalité des commandes allant être utilisées par les clients et le serveur durant leurs échanges. Une fois le protocole établi de manière définitive, celui-ci fera l’objet d’une implémentation sous la forme d’une classe.

* Architecture de base client-serveur

La mise en place de l’architecture de base client-serveur consistera à la gestion des sockets i.e. l’écoute sur un porte, l’attente de clients, la gestion des connexions clientes et une communication basique entre le client et le serveur. Le but de cette partie portera sur le test de l’implémentation du protocole par l’envoi de quelques commandes de celui-ci. Un client et un serveur devront être développés

#### Durée

Une semaine.

#### Partage du travail

Les points suivants constituent le partage du travail pour la première itération :

* Michelle Meguep s’occupera de la partie relative à la base de données.
* Raphaël Racine s’occupera de la partie protocole client-serveur.
* Yassin Kammoun et Guillaume Bruge s’occuperont de l’architecture de base client-serveur.

L’ensemble du groupe s’occupera de la planification de la prochaine itération et de la mise à jour du rapport de travail.

### Itération 2 : 29.04.2015 - 06.05.2015

#### Objectif

La deuxième itération consistera en la manipulation de la base de données du jeu et au développement de l’interface graphique permettant de la manipuler.

#### Description

Les points suivants constituent le descriptif détaillé du contenu du travail effectué lors de la deuxième itération:

* Edition des cartes

L’édition des cartes du jeu devra être rendue possible par le biais d’une interface graphique. Celle-ci devra permettre de créer, de modifier ou de supprimer des cartes. La définition d’une carte consistera en l’upload d’une image de fond, de la définition de stations et de liens entre eux.

* Edition des utilisateurs

L’édition des utilisateurs devra être rendue possible par le biais d’une interface graphique. Celle-ci devra permettre de bloquer des utilisateurs, de rechercher des utilisateurs et de consulter leurs informations et de les supprimer éventuellement.

#### Durée

Une semaine.

#### Partage du travail

Les points suivants constituent le partage du travail pour la deuxième itération :

* Michelle Meguep et Guillaume Bruge s’occuperont de la partie relative à l’édition de comptes utilisateurs.
* Raphaël Racine et Yassin Kammoun s’occuperont de la partie relative à l’édition de cartes du jeu.

L’ensemble du groupe s’occupera de la planification de la prochaine itération et de la mise à jour du rapport de travail.

### Itération 3 : 06.05.2015 - 13.05.2015

#### Objectif

La troisième itération consistera en la mise en place de la gestion des comptes utilisateurs et au développement de l’interface graphique de salle de parties.

#### Description

Les points suivants constituent le descriptif détaillé du contenu du travail effectué lors de la troisième itération:

* Connexion/Déconnexion

La connexion et déconnexion d’un utilisateur par le biais d’un compte utilisateur devra être rendue possible. Celle-ci nécessitera le développement d’une interface graphique de connexion. Elle constituera l’interface d’accueil à l’utilisateur de l’application cliente.

* Edition de compte

L’édition de compte utilisateur devra être possible. Il s’agira de permettre à un client de créer un compte ou d’éditer un compte i.e. de modifier son mot de passe. Deux interfaces graphiques devront être conçues et développées pour ce faire sous la forme de formulaires. Une politique de nom d’utilisateur et de mot de passe devra être établie. Celle-ci devra être appliquée lors d’une tentative de création de compte.

* Interface de salle de partie

Une interface graphique de salle de partie devra être développée, celle-ci devant permettre de disposer d’une liste de parties, d’en créer ou d’en rejoindre. Cependant, l’itération courante portera uniquement sur la conception et le développement de l’interface. Les événements liés à celle-ci seront implémentés lors de l’itération suivante.

#### Durée

Une semaine.

#### Partage du travail

Les points suivants constituent le partage du travail pour la troisième itération :

* Raphaël Racine s’occupera de la partie connexion/déconnexion de compte utilisateur.
* Michelle Meguep et Guillaume Bruge s’occuperont de la partie gestion de compte utilisateur.
* Yassin Kammoun s’occupera d la partie interface graphique de salle de partie.

L’ensemble du groupe s’occupera de la planification de la prochaine itération et de la mise à jour du rapport de travail.

### Itération 4 : 13.05.2015 - 20.05.2015

#### Objectif

La quatrième itération consistera en la mise en place de la gestion et l’édition de parties sans entrer dans le jeu à proprement parler.

#### Description

Les points suivants constituent le descriptif détaillé du contenu du travail effectué lors de la quatrième itération :

* Créer/annuler une partie

L’interface graphique de la salle de parties mise en place lors de l’itération précédente devra permettre de créer une partie. La création d’une partie sera rendue possible par la mise à disposition d’une interface graphique conçue et implémentée à cet effet. Celle-ci devra permettre entre autres de définir le nombre de jours de la partie et de sélectionner une carte de jeu. Il devra être veillé à ce que le nombre de parties pouvant exister simultanément soit contrôlé. L’utilisateur pourra également annuler la partie qu’il tentera de la créer, la supprimer autrement dit.

* Rejoindre une partie

Rejoindre une partie devra être possible au travers de l’interface graphique de salles de partie. Il devra être veillé à ce que rejoindre une partie soit possible i.e. qu’un nombre de joueurs soit insuffisant et que la partie n’a pas débuté. Dans une telle situation, l’utilisateur devra être dirigé vers une interface graphique faisant office de pré-jeu.

* Lancer une partie

Le lancement d’une partie consistera en la phase de pré-jeu, d’initialisation autrement dit. L’utilisateur hôte attendra un nombre d’utilisateurs suffisant pour jouer. Il procédera ensuite au déclenchement du début des votes devant « élire » Mister X. Une fois cette donnée du jeu fixée, il pourra lancer le jeu à proprement dit.

* Gestion des parties

Le serveur devra pour sa part être en mesure de manière totalement fonctionnelle chaque partie, de leur attribuer un thread faisant office de gestionnaire de partie et de veiller à la bonne création d’une partie.

#### Durée

Une semaine.

#### Partage du travail

Les points suivants constituent le partage du travail pour la quatrième itération :

* Michelle Meguep s’occupera de la partie devant permettre de rejoindre une partie.
* Guillaume Bruge s’occupera de la gestion des parties.
* Raphaël Racine s’occupera de la partie relative au lancement d’une partie.
* Yassin Kammoun s’occupera de la partie relative à la création et suppression d’une partie.

L’ensemble du groupe s’occupera de la planification de la prochaine itération et de la mise à jour du rapport de travail.

### Itération 5 : 20.05.2015 - 27.05.2015

#### Objectif

Pour les 3 prochaines itérations (5, 6 et 7), avons décidé de découper en 3 semaines la phase de développement du jeu en lui-même qui sont les suivantes :

* Mise en place des interfaces graphiques, sans que le joueur puisse jouer pour l’instant (itération 5).
* Gestion du déplacement des joueurs sur la carte et affichage des pions sur le plateau de jeu (Itération 6).
* Application des règles du jeu et des coups spéciaux de Mister X (itération 7).

#### Description

Les points suivants constituent le descriptif détaillé du contenu du travail effectué lors de la cinquième itération :

* Disputer une partie
  + Jouer un tour
  + Jouer en tant que détective
  + Jouer en tant que Mister X
  + Se déplacer
  + Choisir un moyen de transport

#### Durée

Une semaine.

#### Partage du travail

Les points suivants constituent le partage du travail pour la cinquième itération :

* Raphaël Racine et Guillaume Bruge s’occuperont de la partie de l’interface graphique qui affiche la carte du jeu (pour un détective et pour un joueur)
* Yassin Kammoun et Michelle Meguep s’occuperont de l’affichage de l’état des tickets des détectives et de l’affichage du tableau des tours avec les déplacements de Mister X.

### Itération 6 : 27.05.2015 - 03.06.2015

#### Objectif

L’objectif de cette itération est expliqué dans l’itération 5.

#### Description

Les points suivants constituent le descriptif détaillé du contenu du travail effectué lors de la sixième itération :

* Disputer une partie
  + Jouer un tour
  + Jouer en tant que détective
  + Jouer en tant que Mister X
  + Se déplacer
  + Choisir un moyen de transport

#### Partage du travail

Les points suivants constituent le partage du travail pour la sixième itération :

* Guillaume Bruge et Michelle Meguep s’occuperont d’afficher les pions des joueurs à chaque fois qu’un joueur s’est déplacé.
* Raphaël Racine et Yassin Kammoun seront chargée de développer la partie qui concerne le déplacement d’un joueur.

### Itération 7 : 03.06.2015 - 10.06.2015

#### Objectif

L’objectif de cette itération est expliqué dans l’itération 5.

#### Description

Les points suivants constituent le descriptif détaillé du contenu du travail effectué lors de la septième itération:

* Disputer une partie
  + Jouer un tour
  + Jouer en tant que détective
  + Jouer en tant que Mister X
  + Se déplacer
  + Choisir un moyen de transport

#### Durée

Une semaine.

#### Partage du travail

Les points suivants constituent le partage du travail pour la septième itération :

* Raphaël Racine et Michelle Meguep devront implémenter les algorithmes de vérifications des déplacements des joueurs dans la partie client
* Yassin Kammoun et Guillaume Bruge devront implémenter les coups spéciaux de Mister X.

### Itération 8 : 10.06.2015 - 17.06.2015

#### Objectif

* Arrêter une partie
* Quitter une partie
* Tchater
* Consulter Statistiques

#### Durée

Une semaine.

#### Partage du travail

Les points suivants constituent le partage du travail pour la huitième itération :

* Raphael Racine et Yassin Kammoun seront chargés de développer le tchat dans une partie
* Guillaume Bruge et Michelle Meguep seront chargés de développer la consultation des statistiques et le fait de pouvoir arrêter une partie ou quitter une partie.

## Plan d’itérations finale

### Itération 1 : 22.04.2015 - 29.04.2015

L’itération portera sur une semaine.

#### Base de données

##### Ce qui sera fait

La base de données du jeu sera créée conformément au modèle conceptuel établi. Cela consistera à définir et créer les tables pour chaque entité du modèle et celles issues des relations entre elles. Le format des données sera également défini lors de cette itération.

*À noter que ce qui sera fait ne correspond à aucun cas d’utilisation. Cette itération fait office de mise en place des bases du projet.*

##### Ce qui sera testé

Les tests que nous réaliserons porteront sur les procédures stockées et les contraintes d’intégrité que nous aurons définies au cours de cette itération. Celles-ci devront restées inchangées pour la suite du projet. Il s’agira de vérifier le caractère fonctionnel des procédures stockées et des contraintes d’intégrité par l’exécution de requêtes. Ces requêtes consisteront en des exemples d’insertion de données valides et de données invalides. Vu que les tables de la base de données seront remplies au fur et à mesure de la création des comptes d’utilisateurs et des parties disputées, nous remplirons dans un premier temps la base de données manuellement avec des données arbitraires.

##### Ce qui sera observé

On pourra observer toutes les tables qui constitueront notre base de données et leur contenu. On pourra également disposer de vues sur les statistiques relatives à un joueur, les informations d’un compte utilisateur et différentes cartes fictives du jeu. Des requêtes seront également exécutées de manière à montrer des insertions correctes et des insertions incorrectes ne respectant pas les contraintes d’intégrité.

#### Architecture de base client-serveur et protocole

##### Ce qui sera fait

Il s’agira de concevoir dans un premier temps l’architecture de base client-serveur c’est-à-dire de mettre en place la communication entre client et serveur.

Cette base est importante pour préparer le terrain des itérations futures qui s’y appuieront conformément au protocole de communication. Ce protocole dont il est question sera complété au fur et à mesure des itérations.

Il s'agira ensuite de définir la signature de la plupart des méthodes qui seront invoquées lorsque le serveur reçoit une commande de la part d'un client. Il en ira de même dans l’autre sens.

Cela va impliquer de définir les classes principales du programme comme par exemple Client, Serveur, Partie (dont la plupart des méthodes seront vides et se complèteront au fur et à mesure des itérations). Cela nous permettra de tester le comportement du serveur ou du client lors de réception de commandes correspondant au protocole de communication défini.

*À noter que ce qui sera fait ne correspond à aucun cas d’utilisation. Cette itération fait office de mise en place des bases du projet.*

##### Ce qui sera testé

Ce qui sera testé sera principalement les connexions entre le client et le serveur ; concrètement, le serveur devra être en mesure d’accepter la connexion de plusieurs clients et d’envoyer un message à l’un ou plusieurs de ces derniers et également en recevoir.

Ensuite, sera également testé le comportement du serveur ou du client lorsque ceux-ci devront invoquer des méthodes qui correspondent à la réception de commandes définies dans le protocole.

Ceci se fera par le biais de plusieurs connexions en ligne de commande avec telnet et ce qui pourra être vu sera simplement au départ des « System.out.println » indiquant bien quelle méthode aura été appelée suite à la réception d’un message par le client ou le serveur, à travers l’envoi d’une commande quelconque dans Telnet parmi la liste suivante :

* PLAY\_DETECTIVE\_TURN
* PLAY\_MISTER\_X TURN
* NOTIFY\_DETECTIVE\_PLAYED
* NOTIFY\_MISTER\_X\_PLAYED
* LAUNCH\_GAME
* ACCEPT\_CREATION\_GAME
* REJECT\_CREATION\_GAME

Le comportement attendu des commandes sera évidemment pour l’instant d’afficher un message dans la console pour montrer que c’est la bonne méthode qui a été appelée ; il ne s’agira évidemment pas du comportement final qui sera plus complexe.

##### Ce qui sera observé

Les résultats de différents envois de commande entre le client et le serveur par le biais du programme Telnet d’une invite de commande (voir rubrique « Ce qui sera testé » pour plus d’informations).

Sur le serveur, il sera possible d’observer par exemple le nombre de clients actuellement connecté en envoyant simplement une commande interne (après connexion et déconnexion de plusieurs clients…) qui seront simulées à travers le programme Telnet.

### Itération 2 : 29.04.2015 - 06.05.2015

L’itération portera sur une semaine.

#### Communication serveur - base de données

##### Ce qui sera fait

L’itération consistera en la mise en place de la communication entre l’application serveur et la base de données locale. L’objectif est de permettre au serveur de pouvoir interroger la base de données par le biais de requêtes.

*À noter que ce qui sera fait ne correspond à aucun cas d’utilisation.*

##### Ce qui sera testé

Il s’agira de tester depuis le serveur la connexion à la base de données à l’aide du Framework JDBC.

Le serveur devra être capable de transmettre une requête à la base de données et d’attendre éventuellement un résultat par exemple lors d’une requête de sélection de données.

##### Ce qui sera observé

On pourra observer les requêtes émises par le serveur à la base de données via la terminal ainsi que les réponses de cette dernière. De plus, les résultats devront être visibles dans les interfaces graphiques développées.

#### Edition des comptes utilisateurs

##### Ce qui sera fait

Cas d’utilisation :

* Bloquer un utilisateur.
* Supprimer un utilisateur.

Cette itération fera également l’objet de la définition de règles de nom d’utilisateur et de mot de passe. De plus, des interfaces graphiques répondant aux cas d’utilisations mentionnés seront développées.

##### Ce qui sera testé

Les tests porteront sur le bon respect des contraintes d’intégrité liées à l’édition de comptes utilisateurs d’une part. Il s’agira également d’autre part de tester le bon fonctionnement des fonctionnalités correspondant aux cas d’utilisation.

##### Ce qui sera observé

Une interface graphique proposant des fenêtres permettant de :

* Lister les utilisateurs.
* Rechercher un utilisateur.
* Ajouter un utilisateur.
* Bloquer un utilisateur.
* Supprimer un utilisateur.

#### Edition des cartes

##### Ce qui sera fait

* Cas d’utilisation :
* Ajouter une carte.
* Supprimer une carte

Des interfaces graphiques répondant aux cas d’utilisations mentionnés seront développées.

##### Ce qui sera testé

Les tests porteront sur le bon respect des contraintes d’intégrité liées à l’édition d’une carte d’une part. Il s’agira également d’autre part de tester le bon fonctionnement des fonctionnalités correspondant aux cas d’utilisation.

##### Ce qui sera observé

Une interface graphique proposant des fenêtres permettant de :

* Lister les cartes.
* Rechercher une carte.
* Ajouter une carte.
  + Définir un arrière-plan (à partir d’un fichier en local).
  + Ajouter des stations (définir des types de station).
  + Ajouter des liens entre stations (définir des moyens de déplacement).
* Supprimer une carte.

### Itération 3 : 06.05.2015 - 13.05.2015

L’itération portera sur une semaine.

#### Edition des cartes

Cette partie initialement prévue lors de l’itération 2 a été reportée à l’itération 3.

##### Ce qui sera fait

Cas d’utilisation :

* Ajouter une carte.
* Supprimer une carte

Des interfaces graphiques répondant aux cas d’utilisations mentionnés seront développées.

##### Ce qui sera testé

Les tests porteront sur le bon respect des contraintes d’intégrité liées à l’édition d’une carte d’une part. Il s’agira également d’autre part de tester le bon fonctionnement des fonctionnalités correspondant aux cas d’utilisation.

##### Ce qui sera observé

Une interface graphique proposant des fenêtres permettant de :

* Lister les cartes.
* Rechercher une carte.
* Ajouter une carte.
  + Définir un arrière-plan (à partir d’un fichier en local).
  + Ajouter des stations (définir des types de station).
  + Ajouter des liens entre stations (définir des moyens de déplacement).
* Supprimer une carte.

#### Edition des comptes utilisateurs

Cette partie n’ayant pas été terminée lors de l’itération 2 a été reportée à l’itération 3.

##### Ce qui sera fait

Cas d’utilisation :

* Bloquer un utilisateur.
* Supprimer un utilisateur.

##### Ce qui sera testé

Les tests porteront sur le bon fonctionnement des fonctionnalités correspondant aux cas d’utilisation i.e. la suppression d’un utilisateur non connecté ou le blocage d’un utilisateur.

##### Ce qui sera observé

Une interface graphique permettant de :

* Bloquer un utilisateur.
* Supprimer un utilisateur.

#### Salle de parties

##### Ce qui sera fait

Une interface graphique de salle de partie sera développée, celle-ci devra permettre de disposer d’une liste de parties, de boutons permettant d’en créer ou d’en rejoindre, de boutons permettant d’éditer son compte et de se déconnecter. Cependant, dans le cadre de cette itération, il s’agira simplement de concevoir et de développer l’interface graphique sans pour autant traiter les événements pouvant s’y produire tels que l’action click sur un bouton ou autres à l’exception de la déconnexion qui sera fonctionnelle.

*À noter que ce qui sera fait ne correspond à aucun cas d’utilisation. Cette itération fait office de mise en place des bases pour l’itération suivante.*

##### Ce qui sera testé

La validation de cette partie consistera en les deux points suivants :

* vérifier que l’interface corresponde graphiquement à la description faite dans « Ce qui sera fait ».
* vérifier qu’un utilisateur ayant réussi à s’authentifier soit redirigé vers cette interface.
* vérifier qu’un utilisateur s’étant déconnecté soit redirigé vers l’interface de connexion.

##### Ce qui sera observé

Une interface graphique disposant :

* d’une zone de listes de parties,
* de boutons permettant de créer ou de rejoindre une partie,
* d’un bouton permettant de se déconnecter,
* d’un bouton permettant d’éditer son compte utilisateur.

#### Connexion, déconnexion & création compte utilisateur

##### Ce qui sera fait

Cas d’utilisation :

* Se connecter au serveur avec un compte utilisateur.
* Créer un compte utilisateur.
* Se déconnecter du compte utilisateur.

##### Ce qui sera testé

Cette partie sera testée des points de vue client et serveur :

* Du point de vue client, il s’agira de :
  + bien être redirigé vers l’interface de salle de partie en cas d’authentification réussie,
  + bien être redirigé vers l’interface de connexion en cas de déconnexion.
* Du point de vue serveur, il s’agira de :
  + bien être notifié de l’authentification réussie d’un client,
  + bien être notifié de la déconnexion d’un client.

##### Ce qui sera observé

Une interface graphique permettant de :

* Se connecter à un serveur avec un compte utilisateur en précisant l’adresse ip et le port du serveur.
* Créer un compte utilisateur en spécifiant le nom d’utilisateur et le mot de passe.
* Se déconnecter du serveur.

### Itération 4 : 13.05.2015 - 20.05.2015

L’itération portera sur une semaine.

#### Gestion de parties côté client

La gestion de parties côté client porte sur la mise en place des fonctionnalités permettant de manipuler une partie sans pour autant entrer dans le jeu à proprement parlé. Il s’agit concrètement de pouvoir disposer de moyens permettant de gérer une partie à sa guise.

##### Ce qui sera fait

Cas d’utilisation :

* Gérer une partie
* Créer une partie
* Supprimer une partie
* Choisir une carte
* Fixer le nombre de joueurs

##### Ce qui sera testé

Les tests porteront sur le bon respect des contraintes d’intégrité liées à la gestion d’une partie Il s’agira également d’autre part de tester le bon fonctionnement des fonctionnalités correspondant aux cas d’utilisation.

##### Ce qui sera observé

Les points suivants décrivent les interfaces et les comportements attendus allant pouvoir être observés :

* Créer/annuler une partie.

L’interface graphique de la salle de parties mise en place lors de l’itération précédente devra permettre de créer une partie. La création d’une partie sera rendue possible par la mise à disposition d’une interface graphique conçue et implémentée à cet effet. Celle-ci devra permettre entre autres de définir le nombre de jours de la partie et de sélectionner une carte de jeu. Il devra être veillé à ce que le nombre de parties pouvant exister simultanément soit contrôlé. L’utilisateur pourra également annuler la partie qu’il tentera de la créer, la supprimer autrement dit. Bien évidemment, dans le cas contraire, il sera attendu de lui qu’il valide la création de la partie.

* Rejoindre une partie.

Rejoindre une partie devra être possible au travers de l’interface graphique de salles de partie. Il devra être veillé à ce que rejoindre une partie soit possible i.e. qu’un nombre de joueurs soit insuffisant et que la partie n’ait pas débuté. Dans une telle situation, l’utilisateur devra être dirigé vers une interface graphique faisant office de pré-jeu.

* Attente de joueurs.

La validation de la création d’une partie aura pour conséquence la redirection de l’utilisateur hôte vers une interface graphique qui servira simplement à attendre que des utilisateurs rejoignent la partie en phase de préparation, le « pré-jeu ». Une fois un nombre de joueurs jugé suffisant par l’hôte, ce dernier pourra lancer les hostilités ; le jeu à proprement parlé débutera.

#### Gestion de parties côté serveur

Cette partie porte sur la mise en place de la gestion de parties côté serveur.

##### Ce qui sera fait

Cas d’utilisation :

* Gérer une partie.
* Créer une partie.
* Supprimer une partie.
* Choisir une carte.
* Fixer le nombre de joueurs.

##### Ce qui sera testé

Le serveur devra être en mesure de gérer chaque partie, de leur attribuer un thread faisant office de gestionnaire de partie et de veiller à la bonne création d’une partie. Concrètement, il s’agira de vérifier le bon suivi du protocole.

##### Ce qui sera observé

Du point de vue du serveur, des messages en ligne de commande relatifs à la gestion de parties permettront de visualiser la mise en place de cette partie. Toutefois, ce qui sera observé pour la gestion de parties côté client constituera « l’observable » la gestion côté client et celle côté serveur sont étroitement liées.

### Itération 5 : 20.05.2015 - 27.05.2015

L’itération durera une semaine. Elle ne correspondra à aucun cas d’utilisation en particulier. Néanmoins, elle peut être considérée comme étant les bases du cas d’utilisation « Disputer une partie ». Concrètement, il s’agira de développer une grande majorité de la partie graphique du jeu et de toute la partie relative à l’initialisation d’une partie de Scotland Yard.

#### Interface graphique de la plateforme de jeu

##### Ce qui sera fait

Ce qui sera fait consistera au développement de l’interface graphique de la plateforme de jeu côté client qui lui permettra de disputer une partie. Celle-ci devra être composée des zones suivantes :

Une zone relative à la carte du jeu :

* Arrière-plan.
* Stations.
* Liens entre stations.

Une zone relative aux données de chaque joueur :

* Pour les détectives
  + Tickets de taxi.
  + Tickets de bus.
  + Tickets de métro.
  + Une vue sur tous les autres détectives.
* Pour Mister X
  + Tickets de taxi.
  + Tickets de bus.
  + Tickets de métro
  + Tickets noirs.
  + Cartes « coup double ».
  + Une vue sur tous les détectives.
* Une zone relative aux informations générales du jeu
  + Tour.
  + Joueur ayant la main.
  + Départ d’un joueur.
  + Coup joué par un détective (ou par Mister X lorsqu’il devra montrer sa position)
  + Moyen de déplacement utilisé par Mister X.
  + Fin de partie.
  + Camp victorieux.
  + Autres informations possibles…

Toutes ces informations faisant office de logs du jeu seront disponibles au fur et à mesure de la réalisation de la suite des itérations. Il s’agit simplement de décrire le contenu que cette zone tâchera de proposer.

Les pions des joueurs ne seront pour l’instant pas visibles et aussi on ne verra que dans cette itération le nombre de tickets que chaque joueur aura au début de la partie.

La carte du jeu doit s’afficher en haut à gauche et prendra une bonne partie de l’écran, on devra y voir toutes les stations ainsi que les liens qui représente les moyens de transports qu’on peut emprunter d’une station A à une station B.

Ce qu’il y aura en plus dans l’interface graphique de Mister X (l’hôte de la partie) sera deux CheckBox qui lui permettront de dire qu’il veut se déplacer deux fois de suite et / ou cacher son moyen de déplacement.

*À noter que ce qui sera fait ne correspond à aucun cas d’utilisation. Cette itération fait office de mise en place des bases de la prochaine itération.*

##### Ce qui sera testé

Cette partie ne fera l’objet d’aucun test ; il s’agira simplement de vérifier que l’interface graphique répond à la description faite dans la partie « Ce qui sera fait ».

##### Ce qui sera observé

Une interface graphique répondant à la description précédente.

*À noter que compte tenu des différences entre Mister X et les détectives, l’interface graphique de Mister X aura quelques différences mineures.*

#### Construction de la carte du jeu

##### Ce qui sera fait

Ce qui sera fait consistera en la construction de la carte du jeu. Il s’agira de proposer quelque chose de semblable au plateau original du jeu de société Scotland Yard.

* Génération graphique de la carte
* Affichage des stations (avec différences selon type de station).
* Affichage des liens entre station.
* Affichage de l'arrière-plan.
* Définition des écouteurs d'événement sur les stations.

*À noter que ce qui sera fait ne correspond à aucun cas d’utilisation. Cette itération fait office de mise en place des bases de la prochaine itération.*

##### Ce qui sera testé

Les tests seront divisés en deux parties :

* Rendu de la carte.
  + Il s’agira de veiller à ce que la carte générée dans sa forme finale correspond à la représentation de la carte au sein de la base de données. Les stations devront être mises en évidence de manière adéquate selon leur type (taxi, taxisbus, taxibusmetro). Leur position devra correspondre aux coordonnées définies au sein de la base de données. La représentation des liens entre station fera l’objet d’une définition d’une convention de représentation dans le cadre de l’itération.
* Ecouteurs d’événement.
  + Il s’agira de vérifier que les événements clicks soient possibles sur les stations. Cela sera vérifié par l’affichage de message par le biais de messages Sytem.out.println(…). Ceux-ci permettront de vérifier qu’une station a bien fait l’objet d’une action click.

##### Ce qui sera observé

Ce qui sera observé consistera en la bonne génération de la carte sélectionnée par l’utilisateur :

* Les stations devront être visibles.
* Le type des stations devra être mis en évidence (taxi, taxibus, taxibusmétro).
* Les liens entre stations devront être visibles.
* Une convention de représentation de type de lien entre station devra être établie.
* Le type des liens entre stations devra être mis en évidence.
* Les événements liés aux actions clicks sur les stations devront être opérationnels.

#### Initialisation du jeu

##### Ce qui sera fait

Concernant cette partie, le serveur devra initialiser la partie une fois que l’hôte aura cliqué sur le bouton « Lancer » dans l’interface graphique où il attend qu’un nombre suffisant de joueurs se connecte. Une fois que ceci est réalisé, le serveur devra initialiser la partie ce qui consiste à :

* Charger la carte depuis la base de données et la transmettre en JSon à chaque joueur (en binaire pour l’image de l’arrière-plan)
* Transmettre le nombre de tickets de chaque joueur à tout le monde qu’ils auront au début de la partie, ainsi que les positions de départ. Les positions de départ seront choisies de manière aléatoire, avec la contrainte qu’aucun joueur ne commence la partie sur la même station.
* Le serveur devra également gérer les pions des joueurs ; Ces informations seront stockées en associant à chaque joueur un pion (qui contiendra le numéro de la case où il se trouve), ainsi que le rôle joué par le pion (détective où Mister X) Structure similaire à ce qu’on peut voir dans le modèle de domaine.

Les joueurs de leurs côté, devront se synchroniser avec le serveur pour la réception de toutes ses informations, et adapter leur interface graphique en conséquence.

À noter que ce qui sera fait ne correspond à aucun cas d’utilisation. Cette itération fait office de mise en place des bases de la prochaine itération.

##### Ce qui sera testé

Ce qui sera testé sera la bonne transmission et la bonne réception des informations du serveur aux différents joueurs i.e. détectives et Mister X. Cela sera vérifié d’une part par le biais de l’outil telnet. Cela sera ensuite vérifié au sein de l’interface graphique dès le moment où son développement sera terminé.

##### Ce qui sera observé

Ce qui sera observé sera la bonne réception des données d’initialisation du jeu (tickets, carte, position) côté client. Celles-ci seront visibles dans la zone prévue pour la visualisation des données de jeu de chaque joueur. De plus, ces informations devront être cohérentes, c’est-à-dire qu’elles devront correspondre aux règles relatives à la distribution du matériel. Chaque joueur devra se voir remis du bon nombre de tickets selon les différents types de ticket.

### Itération 6 : 27.05.2015 - 03.06.2015

L’itération portera sur une semaine.

#### Affichage de la carte

Ce point reprend ce qui n’a pas été fait lors de la précédente itération et le complète avec ce qui a été prévu pour l’itération en question.

##### Ce qui sera fait

* Affichage des liens entre les stations.
* Affichage des pions des joueurs.

Aucun de ces points ne correspond à un cas d’utilisation.

##### Ce qui sera testé

Les tests porteront sur les points suivants :

* Affichage des liens entre les stations.
  + Vérifier que les liens soient visibles.
  + Vérifier que le type des liens soit mis en évidence.
* Affichage des pions des joueurs.
  + Vérifier que chaque joueur soit représenté par une couleur sur la carte (sauf Mister X).
  + Vérifier que la couleur associée à un joueur soit indiquée dans les onglets relatifs à chaque joueur.
  + Vérifier que chaque couleur soit unique pour les différents joueurs.

##### Ce qui sera observé

Ce qui sera observé sera :

* L’affichage des liens entre les stations avec mise en évidence du type de lien.
* L’affichage des pions des joueurs devant avoir chacun une couleur différente de manière à identifier à quel jouer un pion appartient.

#### Dispute d’une partie côté client

##### Ce qui sera fait

Cas d’utilisation :

* Disputer une partie
* Jouer un tour
* Jouer en tant que mister x
* Jouer en tant que détective
* Se déplacer
* Choisir un moyen de déplacement

L’implémentation de ces cas d’utilisations sera réalisée du côté client. Autrement dit, lors de l’exécution d’une de ces fonctionnalités par le client, aucune information ne sera transmise au serveur. Il s’agira donc de pouvoir disputer une partie et de jouer un tour i.e. se déplacer à travers la carte en choisissant un moyen de déplacement (un ticket autrement dit).

En effet, il devra être possible pour le joueur de pouvoir se déplacer sur une autre station de la manière suivante :

* Il cliquera sur la station ou il veut aller, et un menu temporaire s’affichera pour qu’il puisse choisir avec quelle moyen de transport il veut se déplacer (taxi, bus, métro) pour autant que ces moyens de déplacement soit possible pour aller à sa destination
* Le joueur ne pourra se rendre qu’à une station voisine de celle où il se trouve actuellement, et qui de plus n’est pas occupée par quelqu’un d’autre (pour les détectives, ils peuvent aller sur la case où se trouve Mister X)

Mister X pourra choisir le moyen de déplacement « ticket noir » ce qui lui permettra par la suite (itération 7) de se déplacer en cachant son moyen de transport.

Donc concrètement, le but de cette partie et de préparer les interactions avec un joueur pour qu’il puisse « préparer » le coup qu’il va jouer. Le déroulement du coup lui-même sera développé à la prochaine itération.

##### Ce qui sera testé

* Menu pour choisir un moyen de déplacement
  + Il sera vérifié que seul les moyens de déplacement disponibles entre une station A et B (A étant celle où le joueur se trouve) seront affichés dans le menu.
  + Pour Mister X, le déplacement « ticket noir », ce qui lui permet de cacher son moyen de déplacement, devra toujours être affichée car il a le droit d’utiliser ce ticket pour substituer n’importe quel ticket normal (taxi, bus, métro).
  + Il sera vérifié que peu importe le moyen de déplacement (caché ou pas) le bouton concernant le moyen de déplacement sera grisé si l’utilisateur n’a plus de ticket pour l’utiliser.
* Lors du choix d’une station de destination, ce menu devra effectivement apparaître uniquement si la case n’est pas occupée par un détective et qu’elle est voisine

##### Ce qui sera observé

Il devra être possible de :

* Se déplacer d’une station à l’autre.
* Choisir un moyen de déplacement.

### Itération 7 : 03.06.2015 - 10.06.2015

L’itération portera sur une semaine.

Cas d’utilisation :

* Disputer une partie
* Jouer un tour
* Jouer en tant que mister x
* Jouer en tant que détective
* Se déplacer
* Choisir un moyen de déplacement

#### Gestion des tours côtés serveur

##### Ce qui sera fait

Au fur et à mesure de l’avancement dans la partie, le serveur devra gérer à qui c’est le tour de jouer. Il tiendra compte du fait que pour certains tours (3ème, 8ème, 13ème et 18ème tour) MisterX doit montrer sa position aux détectives, donc lorsque le serveur transmettra les coups joués, il dira si Mister X doit apparaître ou pas.

Le serveur sera chargé d’envoyer le numéro du tour aux joueurs avec le nom de celui qui doit jouer.

Attention, si Mister X se déplace 2 fois de suite, ça compte comme 2 tours au lieu d’un seul.

##### Ce qui sera testé

Il sera testé que le serveur gère correctement la gestion des tours (en affichant des messages dans la console).

##### Ce qui sera observé

Il sera observé que l’ordre des tours est correct et qu’on peut voir l’apparition de Mister X du point de vue des détectives lors du 3ème , 8ème, 13ème et 18ème tour.

#### Mise à jour des tickets

##### Ce qui sera fait

Le déplacement d’un joueur aura pour conséquence la mise à jour de ses tickets. Dans le cas d’un détective, son ticket sera transmis à Mister X conformément aux règles du jeu. Le ticket consommé par Mister X sera quant à lui définitivement perdu.

Le ticket joué par un joueur sera transmis au serveur. Celui-ci se chargera de communiquer à chaque joueur quel joueur a consommé quel ticket. L’état des tickets de chaque joueur devra être mise à jour en conséquence dans les différents onglets de la plateforme de jeu.

##### Ce qui sera testé

La mise à jour des tickets sera validée de la manière suivante :

* Mister X consomme un ticket.
  + Chaque joueur en est informé.
  + L’onglet relatif aux tickets de Mister X sera réactualisé pour chaque joueur.
* Un détective consomme un ticket.
  + Chaque joueur en est informé.
  + L’onglet relatif aux tickets du détective en question sera réactualisé pour chaque joueur.
  + L’onglet relatif aux tickets de Mister X sera mis à jour par l’ajout du ticket utilisé par le détective.

##### Ce qui sera observé

Lors d’un tour joué par un joueur, les changements et événements suivants seront observés :

* L’état des tickets du joueur est mis à jour du côté de chaque joueur.
* La zone de notification de chaque joueur indique quel ticket a été consommé par quel joueur.

#### Détection de fin de partie

##### Ce qui sera fait

La détection de fin de partie sera réalisée durant l’itération. Celle-ci mettra fin à la dispute d’une partie. Elle sera vérifiée lors de chaque tour ; une partie sera considérée comme étant terminée si l’une des conditions suivante est atteinte :

* Mister X a été trouvé.
  + Un joueur se trouve sur la même station que Mister X.
* Dernier tour de Mister X achevé.
  + Mister X a effectué son 24ème déplacement et aucun détective ne l’as trouvé après son déplacement.

À noter qu’il s’agit d’un OU exclusif ; soit Mister X a été trouvé, soit il a effectué son 24ème déplacement.

##### Ce qui sera testé

La validation de la détection de fin de partie consistera à s’assurer que les règles de fin de partie soient respectées ; c’est-à-dire qu’une fin de partie soit bel et bien détectée lorsque Mister X a été trouvé ou lorsque celui-ci a procédé à son 24ème déplacement et que aucun détective ne l’a trouvé après son déplacement.

##### Ce qui sera observé

La fin d’une partie sera notifiée dans la zone de notifications d’événements ; les informations suivantes pourront être constatées :

* Affichage du message « Fin de partie ».
* Affichage du message « Victoire de MisterX/l’équipe des détectives ».

#### Zone de notification d’événements

##### Ce qui sera fait

L’interface graphique d’une partie disposant d’une zone s’apparentant à une liste graphique, celle-ci, consistant en une zone de notifications, notifiera les événements se produisant relatifs et durant une partie. Les informations en question sont les suivantes :

* Tour courant.
* Joueur ayant la main.
* Départ d’un joueur.
* Coup joué par un détective (ou par Mister X lorsqu’il devra montrer sa position)
* Moyen de déplacement utilisé par Mister X.
* Fin de partie.

##### Ce qui sera testé

La validation de cette partie de l’itération sera réalisée simplement par la vérification de la bonne réception et donc, le bon affichage des informations énumérées dans la section précédente à savoir :

* Tour courant.
* Joueur ayant la main.
* Départ d’un joueur.
* Coup joué par un détective (ou par Mister X lorsqu’il devra montrer sa position)
* Moyen de déplacement utilisé par Mister X.
* Fin de partie.

##### Ce qui sera observé

Ce qui sera observé sera la bonne mise à jour de la zone de notification d’événements avec les informations suivantes :

* Tour courant.
* Joueur ayant la main.
* Départ d’un joueur.
* Coup joué par un détective (ou par Mister X lorsqu’il devra montrer sa position)
* Moyen de déplacement utilisé par Mister X.
* Fin de partie.

### Itération 8 : 10.06.2015 - 17.06.2015

L’itération portera sur une semaine. Elle consiste d’une part à rattraper le travail de l’itération précédente avec abandon de la notification des événements et événements des particularités relatives à Mister X lorsque celui-ci est appelé à jouer. D’autre part, l’itération consiste à finaliser le projet comprenant rapport et code.

L’itération portera sur une semaine.

Cas d’utilisation :

* Disputer une partie
* Jouer un tour
* Jouer en tant que mister x
* Jouer en tant que détective
* Se déplacer
* Choisir un moyen de déplacement

#### Gestion des tours côtés serveur

##### Ce qui sera fait

Au fur et à mesure de l’avancement dans la partie, le serveur devra gérer à qui c’est le tour de jouer. Il tiendra compte du fait que pour certains tours (3ème, 8ème, 13ème et 18ème tour) MisterX doit montrer sa position aux détectives, donc lorsque le serveur transmettra les coups joués, il dira si Mister X doit apparaître ou pas.

Le serveur sera chargé d’envoyer le numéro du tour aux joueurs avec le nom de celui qui doit jouer.

Attention, si Mister X se déplace 2 fois de suite, ça compte comme 2 tours au lieu d’un seul.

##### Ce qui sera testé

Il sera testé que le serveur gère correctement la gestion des tours (en affichant des messages dans la console).

##### Ce qui sera observé

Il sera observé que l’ordre des tours est correct et qu’on peut voir l’apparition de Mister X du point de vue des détectives lors du 3ème , 8ème, 13ème et 18ème tour.

#### Mise à jour des tickets

##### Ce qui sera fait

Le déplacement d’un joueur aura pour conséquence la mise à jour de ses tickets. Dans le cas d’un détective, son ticket sera transmis à Mister X conformément aux règles du jeu. Le ticket consommé par Mister X sera quant à lui définitivement perdu.

Le ticket joué par un joueur sera transmis au serveur. Celui-ci se chargera de communiquer à chaque joueur quel joueur a consommé quel ticket. L’état des tickets de chaque joueur devra être mise à jour en conséquence dans les différents onglets de la plateforme de jeu.

##### Ce qui sera testé

La mise à jour des tickets sera validée de la manière suivante :

* Mister X consomme un ticket.
  + Chaque joueur en est informé.
  + L’onglet relatif aux tickets de Mister X sera réactualisé pour chaque joueur.
* Un détective consomme un ticket.
  + Chaque joueur en est informé.
  + L’onglet relatif aux tickets du détective en question sera réactualisé pour chaque joueur.
  + L’onglet relatif aux tickets de Mister X sera mis à jour par l’ajout du ticket utilisé par le détective.

##### Ce qui sera observé

Lors d’un tour joué par un joueur, les changements et événements suivants seront observés :

* L’état des tickets du joueur est mis à jour du côté de chaque joueur.
* La zone de notification de chaque joueur indique quel ticket a été consommé par quel joueur.

#### Détection de fin de partie

##### Ce qui sera fait

La détection de fin de partie sera réalisée durant l’itération. Celle-ci mettra fin à la dispute d’une partie. Elle sera vérifiée lors de chaque tour ; une partie sera considérée comme étant terminée si l’une des conditions suivante est atteinte :

* Mister X a été trouvé.
  + Un joueur se trouve sur la même station que Mister X.
* Dernier tour de Mister X achevé.
  + Mister X a effectué son 24ème déplacement et aucun détective ne l’as trouvé après son déplacement.

À noter qu’il s’agit d’un OU exclusif ; soit Mister X a été trouvé, soit il a effectué son 24ème déplacement.

##### Ce qui sera testé

La validation de la détection de fin de partie consistera à s’assurer que les règles de fin de partie soient respectées ; c’est-à-dire qu’une fin de partie soit bel et bien détectée lorsque Mister X a été trouvé ou lorsque celui-ci a procédé à son 24ème déplacement et que aucun détective ne l’a trouvé après son déplacement.

##### Ce qui sera observé

La fin d’une partie sera notifiée dans la zone de notifications d’événements ; les informations suivantes pourront être constatées :

* Affichage du message « Fin de partie ».
* Affichage du message « Victoire de MisterX/l’équipe des détectives ».

#### Zone de notification d’événements

##### Ce qui sera fait

L’interface graphique d’une partie disposant d’une zone s’apparentant à une liste graphique, celle-ci, consistant en une zone de notifications, notifiera les événements se produisant relatifs et durant une partie. Les informations en question sont les suivantes :

* Tour courant.
* Joueur ayant la main.
* Départ d’un joueur.
* Coup joué par un détective (ou par Mister X lorsqu’il devra montrer sa position)
* Moyen de déplacement utilisé par Mister X.
* Fin de partie.

##### Ce qui sera testé

La validation de cette partie de l’itération sera réalisée simplement par la vérification de la bonne réception et donc, le bon affichage des informations énumérées dans la section précédente à savoir :

* Tour courant.
* Joueur ayant la main.
* Départ d’un joueur.
* Coup joué par un détective (ou par Mister X lorsqu’il devra montrer sa position)
* Moyen de déplacement utilisé par Mister X.
* Fin de partie.

##### Ce qui sera observé

Ce qui sera observé sera la bonne mise à jour de la zone de notification d’événements avec les informations suivantes :

* Tour courant.
* Joueur ayant la main.
* Départ d’un joueur.
* Coup joué par un détective (ou par Mister X lorsqu’il devra montrer sa position)
* Moyen de déplacement utilisé par Mister X.
* Fin de partie.

#### Finalisation du rapport

##### Ce qui sera fait

* Ajout des diagrammes de séquence et d’activités.
* Mise au propre.
* Relecture.
* Impression.

#### Finalisation du jeu

##### Ce qui sera fait

* Mise au propre.
* Factorisation.
* Commentaire.

#### Edition des cartes (optionnel)

Cette partie porte sur la finalisation de l’application côté administrateur du serveur. Il s’agit de permettre à tout administrateur d’éditer à sa guise une carte ayant été créée au préalable et désormais disponible sur le serveur.

##### Ce qui sera fait

Cas d’utilisation :

* Éditer une carte.
  + Station
    - Modifier le numéro.
    - Modifier les coordonnées.
    - Modifier le type.
  + Lien
    - Changer les extrémités.
    - Modifier le type.

##### Ce qui sera testé

Les tests porteront sur le bon respect des contraintes d’intégrité liées à la modification d’une carte d’une part, les contraintes étant les mêmes que celles mises en place pour l’édition d’une carte (ajout de station/lien). Il s’agira également d’autre part de tester le bon fonctionnement des fonctionnalités correspondant aux cas d’utilisation.

##### Ce qui sera observé

Une interface graphique proposant des fenêtres permettant de :

* Modifier une station.
* Modifier un lien.

## Suivi de projet

### Itération 1 : 22.04.2015 - 29.04.2015

#### Bilan global

##### Objectifs

Les objectifs fixés pour cette itération ont tous été atteints sans exception.

##### Planning

Le planning préliminaire établi pour l’ensemble des itérations n’a pas été modifié étant donné que le travail réalisé durant l’itération n’a généré aucun retard.

##### Difficultés globales

L’unique difficulté rencontrée lors de la réalisation de cette itération est la non-participation de Guillaume Bruge. Celui-ci ayant été injoignable durant toute une semaine pour des raisons que ne nous connaissons pas, son travail a dû être réparti entre les membres du groupe restants.

#### Bilan Guillaume Bruge

##### Travail effectué

Aucun

##### Effort consacré

0 heures

##### Difficultés rencontrées

Malade

#### Bilan Yassin Kammoun

##### Travail effectué

* Mise à jour du rapport intermédiaire suite aux remarques et demandes de modifications des évaluateurs.
* Développement d’un premier jet de l’application cliente ; celle-ci permettant dans l’état de cette itération de se connecter au serveur et de lui envoyer un message.
* Test de la connexion d’un client au serveur et envoie d’une commande bidon.

##### Effort consacré

3-4 heures.

##### Difficultés rencontrées

Aucune difficulté n’a été rencontrée à l’exception de la description textuelle d’une relation de type extends lors de la définition des scénarios.

#### Bilan Michelle Meguep

##### Travail effectué

Ma mission dans la première itération de ce projet a portée sur la création de la base de données de notre application. Il s’agissait de se baser sur le modèle conceptuel établi pour créer des tables, établir des contraintes d’intégrités, et procédures stockées qui régiront le fonctionnement de la base.

Globalement, l’objectif pour cette itération a été atteint. La seule tâche annoncée qui n’a pas été réalisée est l’écriture des procédures stockées .Il a été décidé avec les autres membres du groupe que celles-ci seront gérées et ajoutées au fur et à mesure de l’évolution dans le projet. Ce qui semble le plus logique.

##### Effort consacré

La mise sur pied de la base de donnée n’a pas sollicité beaucoup d’effort et de temps.4 heures ont été suffisantes. Ce qui semble raisonnable au vu de la petitesse de notre base de données.

##### Difficultés rencontrées

Aucun problème majeur n’a été rencontré à cette itération.

#### Bilan Raphaël Racine

##### Travail effectué

* Ebauche du programme serveur avec possibilité d'accepter plusieurs clients et de dialoguer avec eux (effort consacré : 1 h 30)
* Mise en place de l'analyse syntaxique des commandes reçue par un client et effectuer le « bon traitement » (autrement dit appel de la bonne procédure) par rapport à cette commande (1 h)
* Faire quelques tests avec telnet et simuler différentes commandes envoyées depuis le client vers le serveur, avec plusieurs clients connectés (30 min)

Les résultats obtenus ont correspondu avec ceux attendus.

##### Effort consacré

3 heures au total

##### Difficultés rencontrées

Aucun.

### Itération 2 : 29.04.2015 - 06.05.2015

#### Bilan global

##### Objectifs

Les objectifs ayant été atteints lors de l’itération sont les suivants :

* Edition des comptes utilisateurs
  + Lister les utilisateurs.
  + Rechercher un utilisateur.

Les objectifs n’ayant pas été atteints lors de l’itération sont les suivants :

* Edition des comptes utilisateurs
  + Bloquer un utilisateur.
  + Supprimer un utilisateur.
* Edition des cartes
  + Lister les cartes.
  + Rechercher une carte.
  + Ajouter une carte.
  + Supprimer une carte.

##### Planning

Le planning général du plan d’itérations a été revu pour causes :

* De la nouvelle absence de Guillaume Bruge.
* De la non-implémentation des cas d’utilisations relatifs à l’édition de cartes liées à l’absence citée de notre collègue.
* De la non-finalisation des cas d’utilisations relatifs à l’édition des comptes utilisateurs.

En conséquences des points précédents, les mesures suivantes ont été prises :

* L’implémentation des cas d’utilisations relatifs à l’édition de cartes est reportée à l’itération 3.
* La finalisation de l’implémentation des cas d’utilisations relatifs à l’édition des comptes utilisateurs est reportée à l’itération 3.
* Les cas d’utilisations « Tchater », « Désigner Mister X », « Voter Mister X » et « Consulter ses statistiques de jeu » sont abandonnés.

La réalisation du jeu à proprement dit prévue sur les trois itérations 5, 6 et 7 est prolongée sur une itération supplémentaire i.e. 5, 6, 7 et 8.

##### Difficultés globales

La difficulté principale rencontrée lors de l’itération s’est portée sur la gestion des conséquences de l’absence de notre collègue Guillaume Bruge. Il s’est avéré difficile de respecter les délais et d’atteindre les objectifs fixés pour l’itération. En conséquence, l’itération suivante nécessitera un travail plus conséquent afin de se remettre dans les temps.

#### Bilan Guillaume Bruge

##### Travail effectué

Aucun

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Malade

#### Bilan Yassin Kammoun

##### Travail effectué

* Développement de l’interface graphique de gestion des comptes utilisateurs côté administrateur du serveur.
* Définition des règles de définition de nom d’utilisateur et de mot de passe.
* Développement des interfaces graphiques d’ajout et d’édition de comptes utilisateurs côté administrateur du serveur.

##### Effort consacré

3-4 heures.

##### Difficultés rencontrées

Aucune difficulté n’a été rencontrée.

#### Bilan Michelle Meguep

##### Travail effectué

Ma mission dans cette itération porte sur l’édition des cartes de jeu. Il s’agit entre autre de permettre l’ajout ou la suppression d’une carte, d’ajouter des stations et les liens entre elles.

Cette tâche a été reportée à l’itération suivante.

L’édition des cartes ayant été reportée à l’itération suivante, j’ai débuté la mise sur pied de l’interface graphique nécessaire à la création des comptes utilisateurs. Il s’agit de permettre à un utilisateur de créer un compte, de se connecter à un compte et y modifier ses informations personnelles, de rejoindre une partie, de consulter ses statistiques.

##### Effort consacré

6 heures.

##### Difficultés rencontrées

Aucun problème majeur n’a été rencontré à cette itération.

#### Bilan Raphaël Racine

##### Travail effectué

* Connexion de la base de données avec le serveur en utilisant le Framework JDBC
* Adaptation de l'interface graphique de Monsieur Yassin Kammoun sur les différents opérations et événements interagissant avec la base de données (recherche d'utilisateur, liste, ajout, édition etc.)

##### Effort consacré

2-3 heures.

##### Difficultés rencontrées

Aucune difficulté n'a été rencontrée.

### Itération 3 : 06.05.2015 - 13.05.2015

#### Bilan global

##### Objectifs

Les objectifs fixés pour cette itération ont tous été atteints sans exception.

##### Planning

Le planning modifié lors de la précédente itération a bien été suivi et n’a généré aucun retard. Le retard accumulé lors de la deuxième itération a bien été rattrapé.

##### Difficultés globales

Cette itération a fait l’objet de quelques problèmes de communication au sein du groupe. Ceux-ci ont néanmoins été réglés après une mise au point et une discussion avec l’ensemble des protagonistes.

#### Bilan Guillaume Bruge

##### Travail effectué

Aucun.

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Malade.

#### Bilan Yassin Kammoun

##### Travail effectué

* Edition des cartes.
  + Développement de l’interface graphique de la gestion des cartes.
  + Développement de l’interface graphique de la création de cartes.
  + Développement des requêtes SQL relatives à la manipulation de cartes.
  + Implémentation des contraintes d’intégrité liées à la création de cartes.

##### Effort consacré

4-5 heures.

##### Difficultés rencontrées

Aucune difficulté ne s’est présentée.

#### Bilan Michelle Meguep

##### Travail effectué

Créer Une interface graphique de la salle de partie, permettant de voir la liste des parties, des boutons permettant d’en créer ou d’en rejoindre, des boutons permettant d’éditer son compte et de se déconnecter. Cependant, dans le cadre de cette itération, il s’agit simplement de concevoir et de développer l’interface graphique sans pour autant traiter les évènements pouvant s’y produire tels que l’action click sur un bouton ou autres à l’exception de la déconnexion qui sera fonctionnelle.

##### Effort consacré

3 heures.

##### Difficultés rencontrées

Choisir le layout adéquat.

#### Bilan Raphaël Racine

##### Travail effectué

Implémentation côté serveur permettant :

* Connexion d’un utilisateur avec son compte
* Déconnexion d’un utilisateur avec son compte
* Possibilité de bloquer / supprimer un utilisateur

Implémentation côté client :

* Fenêtre de connexion d’un utilisateur sur son compte (adresse IP, port, nom d’utilisateur, mot de passe)
* Possibilité de créer un utilisateur, d’éditer son compte utilisateur, et de se connecter / déconnecter sur son compte

##### Effort consacré

*3 – 4 heures*

##### Difficultés rencontrées

Aucune difficulté n’a été rencontrée.

### Itération 4 : 13.05.2015 - 20.05.2015

#### Bilan global

##### Objectifs

Les objectifs ont tous été atteints sans exception. Les cas d’utilisation ont été implémentés. Ceux-ci se sont révélés tout à fait fonctionnels avec toutefois une exception. La suppression d’une partie et le départ d’un joueur ne sont pas encore fonctionnels.

##### Planning

Malgré la présence d’un bug dans la solution à la fin de l’itération, le planning n’est pas modifié. Le bug en question pourra être considéré comme étant corrigé dès la prochaine itération.

##### Difficultés globales

La première difficulté rencontrée durant l’itération a porté sur la compréhension d’un bug et à sa tentative de correction. La seconde difficulté a consisté à terminer le travail prévu dans les délais fixés. Le fait d’avoir d’autres obligations pour d’autre cours a ralenti la réalisation du travail sans pour autant engendré un retard.

#### Bilan Guillaume Bruge

##### Travail effectué

Aucun.

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Malade.

#### Bilan Yassin Kammoun

##### Travail effectué

* Développement de l’interface graphique de pré-jeu de l’hôte ayant créé une partie.
  + Développement de la communication joueur vers serveur.
    - Départ d’un joueur de la partie.
    - Départ de l’hôte de la partie.
    - Démarrage de la partie.
    - Lancement de la partie.
    - Suppression de la partie.
* Développement de l’interface graphique de pré-jeu de joueur ayant rejoint une partie.
  + Développement de la communication joueur vers serveur.
    - Départ du joueur du jeu.
    - Départ de l’hôte de la partie.
    - Lancement de la partie.
  + Mise à jour de la liste des joueurs ayant rejoint la partie.
* Développement du rafraîchissement de la liste des parties dans l’interface de salle de parties.
  + Parsing du message Json correspondant à la liste des parties.
  + Mise à jour de la liste des parties.

##### Effort consacré

6 heures.

##### Difficultés rencontrées

Compréhension de l’origine du bug lors de la suppression/du départ d’une partie.

#### Bilan Michelle Meguep

##### Travail effectué

Implémentation complète du cas d’utilisation « créer une partie » comprenant :

* Créer l’interface graphique permettant à l’utilisateur de créer une partie.
* Choisir le nombre de joueurs.
* Rendre accessible les cartes de jeu disponibles côté serveur à l’utilisateur créant la partie.
* Communication avec le serveur pour qu’il crée une partie.
* Supprimer une partie en cours de création.

##### Temps consacré

3 heures.

##### Difficultés rencontrées

Pas de difficultés majeures rencontrées.

#### Bilan Raphaël Racine

##### Travail effectué

* Implémentation d’un gestionnaire de parties sur le serveur
  + Le serveur gère la création d’une partie, un joueur qui rejoint une partie ou encore un joueur qui quitte une partie en cours d’attente de joueurs
* Le serveur est capable de sérialiser la liste des parties en JSon (afin que les clients puissent voir la liste des parties
* Le serveur est capable de sérialiser la liste des cartes, également en JSon

Concernant la partie client, tout a été réalisé par Mme Meguep et Monsieur Kammoun.

##### Effort consacré

4 – 5 heures.

##### Difficultés rencontrées

Aucune difficulté au niveau serveur, mais difficultés au niveau client, les joueurs n’arrivaient pas quitter une partie, ce problème a été réglé.

### Itération 5 : 20.05.2015 - 27.05.2015

#### Bilan global

##### Objectifs

À une exception près, les objectifs de l’itération ont tous été atteints. En effet, l’interface graphique de la plateforme de jeu a bien été développée. Celle-ci s’est révélée correspondre à la description présentée dans la section 5.1 « Interface graphique ».

L’initialisation du jeu côté serveur a été implémentée. Le transfert des informations de jeu (carte, stations, liens, tickets) est fonctionnel.

L’affichage de la carte est quant à lui partiellement fonctionnel. Il s’agit là de l’objectif qui n’a pas été totalement atteint ; les liens ne sont pas visibles.

Les bugs rencontrés lors de la précédente itération relatifs à la suppression d’une partie et aux départs de joueurs ont été corrigés. Ces fonctionnalités sont désormais fonctionnelles.

##### Planning

Malgré la non-implémentation de l’affichage des liens, le planning n’est pas modifié. L’affichage des liens est quant à lui reporté à la prochaine itération.

##### Difficultés globales

La difficulté majeure de l’itération a été de terminer le travail prévu dans les temps. La raison est que le début du travail de l’itération a commencé avec du retard. Également, des bugs ayant pris du temps à corriger ont retardé l’avancement.

#### Bilan Guillaume Bruge

##### Travail effectué

Aucun.

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Malade.

#### Bilan Yassin Kammoun

##### Travail effectué

* Correction des bugs de l’itération précédente.
* Réception et affichage de la carte.
* Réception et traitement des données d’initialisation du jeu.

Compte tenu de la dépendance entre nos tâches respectives, j’ai travaillé en étroite collaboration avec mon collègue Raphaël Racine.

##### Effort consacré

6 heures.

##### Difficultés rencontrées

La difficulté majeure rencontrée a été de comprendre la non-réception de messages envoyés par le serveur aux clients.

#### Bilan Michelle Meguep

##### Travail effectué

Réalisation de l’interface graphique d’une partie côté client. Il a été question entre autre de créer un espace pour l’affichage des cartes selon que le joueur soit un détective ou Mister X, de définir les composants qui seront nécessaire à Mister X de se déplacer deux fois de suite ou cacher son moyen de déplacement ; de définir la zone d’affichage de la carte et la chargée avec celle envoyée par le serveur, d’afficher les événements survenant dans le jeu

##### Temps consacré

5 heures.

##### Difficultés rencontrées

Pas de grosse difficulté rencontrée.

#### Bilan Raphaël Racine

##### Travail effectué

* Implémentation de l’initialisation d’une partie sur le serveur (envoi de l’arrière-plan de la carte aux joueurs, les informations de la carte ainsi que le nombre de tickets en début de partie)
* En collaboration avec Monsieur Kammoun, j’ai participé au développement de l’affichage des stations sur la carte de jeu ainsi qu’une partie de l’interface graphique des joueurs

##### Effort consacré

5 heures

##### Difficultés rencontrées

Aucune difficulté au niveau serveur, mais difficultés au niveau client, les joueurs ne recevaient pas correctement l’image de fond de la carte, mais ce problème a été réglé.

### Itération 6 : 27.05.2015 - 03.06.2015

#### Bilan global

##### Objectifs

Les objectifs fixés pour cette itération ont tous été atteints.

Le travail consistait dans un premier temps à rattraper ce qui n’avait pas été terminé lors de l’itération précédente. Il s’agissait d’afficher les liens entre les stations et mettre en évidence leur type : taxi, bus métro. Cela a été réalisé ; les liens sont bien visibles et il a été associé au taxi la couleur noir, au bus la couleur bleu et au métro la couleur orange. La stratégie pour afficher les types de lien consiste à les superposer en jouant avec le code de couleur établi.

La partie relative à l’affichage de la carte a été complétée et finalisée par l’affichage du pion respectif de chaque joueur. Ceux-ci sont mis en évidence par l’affichage d’un cercle coloré englobant le cercle représentant une station.

Le travail consistait dans un second temps à développer la dispute d’une partie côté client. Il s’agissait entre autres d’implémenter en partie les cas d’utilisation « Disputer une partie », « Choisir un moyen de déplacement », « Jouer en tant que détective » et « Jouer en tant que Mister X ». Cela consistait concrètement à rendre possible le déplacement d’un pion d’une station à une autre, de choisir un moyen de déplacement et des tester les contraintes relatives au déplacement d’un pion.

##### Planning

L’itération n’ayant engendré aucun retard et le travail ayant été réalisé durant celle-ci ayant totalement été effectué, le planning n’est pas modifié.

##### Remarque

Suite à la remarque de Monsieur Christophe Greppin lors de la présentation du travail effectué durant l’itération, il a été décidé d’ajouter un nouveau cas d’utilisation permettant à un joueur de passer son tour durant une partie.

##### Difficultés globales

La difficulté majeure de l’itération a été de dessiner les types de lien entre les stations. Il avait en effet été prévu initialement d’afficher un trait par type de lien. Il a été constaté que le carte n’en devenait que plus illisible, particulièrement lorsque celle-ci s’avérait relativement remplie. En conséquence, un code de couleur a été défini de manière à réduire le nombre de liens visibles sur une carte.

#### Bilan Guillaume Bruge

##### Travail effectué

Aucun.

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Malade.

#### Bilan Yassin Kammoun

##### Travail effectué

* Développement partiel de la boîte de dialogue permettant à un détective de se déplacer.
* Développement partiel de la boîte de dialogue permettant à Mister X de se déplacer.
* Développement de l’affichage des liens et type de liens entre stations.
* Participation au développement de l’affichage des pions des joueurs.

##### Effort consacré

5 heures.

##### Difficultés rencontrées

Aucune difficulté n’a été rencontrée.

#### Bilan Michelle Meguep

##### Travail effectué

* Développement partiel de la boîte de dialogue permettant à un détective de se déplacer.
* Développement partiel de la boîte de dialogue permettant à Mister X de se déplacer.
* Participation partiel à l’affichage des liens et type de liens entre stations.

##### Temps consacré

2 heures.

##### Difficultés rencontrées

Aucune difficulté n’a été rencontrée

#### Bilan Raphaël Racine

##### Travail effectué

* Définition de la stratégie d’affichage des types de lien (en collaboration avec Yassin Kammoun et Vanessa Michelle Meguep)
* Afficher les couleurs des joueurs dans les onglets
* Afficher l’emplacement des pions sur la carte
* Elaboration de classes permettant de gérer les informations des pions du côté du client (avec un modèle Observable-Observé pour la mise à jour des tickets et des déplacements des pions)
* En collaboration avec Vanessa Michelle Meguep, mise au point de l’algorithme permettant à un joueur de se déplacer d’un point à l’autre lorsqu’il doit jouer

##### Effort consacré

Environ 5 heures.

##### Difficultés rencontrées

Aucune difficulté n’a été rencontrée

### Itération 7 : 03.06.2015 - 10.06.2015

#### Bilan global

##### Objectifs

Le projet n’a pas fait l’objet d’un avancement en raison du surplus de travail durant la semaine où l’itération devait être réalisée. Les objectifs de l’itération n’ont donc pas été atteints.

##### Planning

Le travail non-fait est reporté à l’itération suivant qui consiste en la dernière étape du projet. Compte tenu du retard engendré par le non-avancement du projet, il a été décidé d’abandonner la partie relative à la notification des événements susceptibles de se produire durant une partie. De plus, si le temps venait à manquer pour la dernière itération, les fonctionnalités caractérisant Mister X à savoir les coups et les tickets spéciaux de Mister X seraient également abandonnées.

##### Difficultés globales

La difficulté de l’itération a consisté à trouver le temps d’avancer le projet ce qui n’a pas été le cas dans le cas présent.

#### Bilan Guillaume Bruge

##### Travail effectué

Aucun.

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Malade.

#### Bilan Yassin Kammoun

##### Travail effectué

Néant.

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Néant.

#### Bilan Michelle Meguep

##### Travail effectué

Néant.

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Néant.

#### Bilan Raphaël Racine

##### Travail effectué

Néant.

##### Effort consacré

0 heures.

##### Difficultés rencontrées

Néant.

### Itération 8 : 10.06.2015 - 17.06.2015 (TOUT LE MONDE)

#### Bilan global

##### Objectifs

##### Planning

##### Difficultés globales

#### Bilan Guillaume Bruge

##### Travail effectué

##### Effort consacré

##### Difficultés rencontrées

#### Bilan Yassin Kammoun

##### Travail effectué

##### Effort consacré

##### Difficultés rencontrées

#### Bilan Michelle Meguep

##### Travail effectué

##### Effort consacré

##### Difficultés rencontrées

#### Bilan Raphaël Racine

##### Travail effectué

* Rédaction du rapport concernant ma partie (environ 4h)
* Commenter tout le code (environ 5h)
* Essayer de résoudre le bug rencontré qui nous a empêcher de finir mais sans succès (environ 4h)
* Préparation de la présentation en collaboration avec Yassin Kammoun (environ 30 min)

##### Effort consacré

Environ 15 heures.

##### Difficultés rencontrées

Résolution du bug (découvert seulement dans cette itération), le bug n’est pas résolu à ce jour.

## Stratégie de tests

Nous n’avons pas élaboré de stratégie de test à proprement parlé. Nous avons plutôt adopté une approche « je teste pendant que je code », nous avons donc testé au fur et à mesure de l’avancement du projet, malheureusement, nous n’avons pas de rapport de tests à fournir.

Mais, au fil des itérations, on a pu à chaque fois montrer un résultat satisfaisant au client.

# Etat des lieux

Voici un état des lieux du logiciel à l’heure actuelle. Il est possible de :

## Ce qui fonctionne

Voici les fonctionnalités qui fonctionnent à ce jour :

* Editer une carte du côté serveur (sans mode graphique)
* Se connecter à un compte utilisateur sur le serveur
* Ajouter, éditer, supprimer ou bloquer un utilisateur depuis le serveur
* Ajouter, éditer un compte utilisateur depuis un client (en étant connecté sur son compte)
* Le serveur affiche la liste des utilisateurs et la liste des cartes existantes
* Un utilisateur connecté mais pas dans une partie peut voir la liste des parties en cours
* Un utilisateur peut créer ou rejoindre une partie
* On peut voir la carte de la partie, les stations et l’emplacement des pions

## Ce qu’il resterait à développer

Malheureusement, suite à un bug non négligeable de réception de message, nous n’avons pas pu implémenter la phase de jeu proprement dite (nous avons uniquement réussi à faire l’initialisation d’une partie).

Nous pensons que pour finir l’application, nous aurions eu besoin d’une semaine supplémentaire (donc d’une itération 9) pour corriger le bug dans un premier temps, et ensuite finaliser l’application soit :

* Trouver le bug et le corriger (avec restructuration très probable de la gestion de la communication entre le serveur et le client)
* Gérer les tours du côté serveur
* Gérer le déplacement d’un joueur (informer tous les autres joueurs du déplacement, et gérer les pertes de tickets)
* Gérer la fin de partie
* Affichage des événements de la partie dans la fenêtre du jeu (par exemple l’apparition de Mister X…)

Il est à noter qu’au vu de la grande charge de travail en cette fin de semestre, nous avons décidé de consacrer du temps pour le rapport plutôt que d’essayer de corriger un bug qui est assez difficile à trouver et qui nous a empêchés de finir l’application.

Une semaine de plus aurait été suffisant pour corriger ce bug et finir le logiciel.

On en conclut que le retard par rapport au produit fini est d’une semaine.

Malheureusement, le bug est tellement important qu’il est très difficile à estimer le temps nécessaire à le résoudre, sachant qu’on y a passé pas mal de temps mais sans succès.

# Auto-critique

## Solution technique(YASSIN)

## Gestion de projet(YASSIN)

## Planification(YASSIN)

## Améliorations(YASSIN)

## Critiques personnelles(TOUT LE MONDE)

### Guillaume Bruge

### Yassin Kammoun(YASSIN)

### Michelle Meguep(VANESSA)

### Raphaël Racine

Ce projet m’a apporté pas mal de connaissances, notamment sur l’utilisation du design pattern Etat. J’ai pu le pratiquer sur un projet et pas seulement en exercice, et je me suis rendu compte qu’il n’est pas toujours facile à mettre en œuvre car quand un objet doit changer d’état, il ne faut pas oublier de le préciser dans le code. Ceci peut conduire à des erreurs inexplicables, car ce pattern est très utile mais difficile à comprendre dans le code.

Ensuite, j’ai pu découvrir l’utilisation des sockets dans un projet, et je me suis rendu compte que transmettre des informations n’est pas une tâche de tout repos, surtout lorsqu’il s’agit de dé-sérialiser des objets complexes, ou d’envoyer des images, le plus dur et de synchroniser le client et le serveur.

Pour les problèmes de synchronisation rencontrés dans l’itération 8, j’aurais dû commencer dès le départ du projet à m’inspirer de modèles déjà existant concernant la partie communication, plutôt que d’essayer un modèle que nous avons vu dans un autre cours, mais qui s’est avéré insuffisant dans ce projet. J’aurais dû donc faire un peu plus de recherche concernant ce côté-là, vu que je me suis principalement occupé de la partie serveur (et donc en partie des connexions avec les clients).

Au fil des itérations, j’ai rencontré plusieurs difficultés toujours en rapport avec ces problèmes de synchronisation, et à chaque fois c’était un nouveau défi à relever pour que ça marche. Malheureusement, en empilant de plus en plus les problèmes de synchronisation, on arrive à un stade où on a des bugs incompréhensibles très difficiles à résoudre et malheureusement l’un d’entre eux ne nous a pas permis de terminer l’application à temps.

Pour ma part, je considère donc que le projet est un échec, mais il faut dire que ce n’est vraiment pas facile quand dans un premier temps on a perdu un membre du groupe, et dans un deuxième temps la communication entre les membres n’était pas toujours satisfaisante (voir insuffisante). En plus au vu de la charge de travail à la fin de ce semestre, je n’ai pas pu me concentrer correctement sur les dernières itérations et je trouve ça dommage.

Pour conclure, je suis quand même content d’avoir participé à ce projet car il m’a permis d’apprendre pas mal de choses et de rencontrer des difficultés quasiment réels de ce qui se passe en entreprise. J’aimerais souligner que même si le projet est un échec, on aurait sûrement réussi à terminer le logiciel si on aurait eu au moins 1 semaine de plus pour finir l’itération 8, mais quand on a un bug qu’on ne peut pas négliger et très difficile à résoudre et qu’on doit en plus écrire un rapport, on est vite pris par le temps malheureusement.

# Conclusion (YASSIN)

# Annexes

## Installation(Vanessa)

## Utilisation(VANESSA)