GEN

Mini projet : Yahtzee

Fabien Franchini

Ibrahim Ounon

Kevin Ponce

Madolyne Dupraz

Rosanne Combremont

# Fonctionnement général de l'applicatif

## Objectifs de base

Dans le cadre du cours de Génie Logiciel (GEN), nous avons pour projet, de concevoir une application client-serveur en adoptant la méthodologie de travail, Unified Process (UP). Nous avons choisi de réaliser une version digitale du Yahtzee, célèbre jeu de société.

### Utilisation de l'applicatif

Le client est accueilli sur une fenêtre d'authentification où il a la possibilité de spécifier l'adresse d'un serveur distant ainsi que ses identifiants, pour s'y connecter. Ensuite, il est redirigé sur la fenêtre principale du serveur. Pour jouer, il peut rejoindre une partie en attente de joueurs ou, en créer une nouvelle. Le cas échéant il doit spécifier le nombre de participants (2 au minimum) et leur temps à disposition pour chaque tour.

Le client a la possibilité de modifier son profil (mot de passe, avatar, description). Le serveur quant à lui dispose d'outils d'administrations permettant, d'ajouter, de modifier ou de supprimer un joueur. Il peut aussi configurer le moteur de jeu et ainsi permettre, par exemple, la création de partie composée de N joueurs.

L'interface de jeu permet d'effectuer les actions permises, de visualiser sa progression (et celle de ses adversaires) et d'obtenir un résumé des combinaisons gagnantes. Il est prévu aussi de mettre à disposition un système permettant de communiquer lors d'une partie. Le joueur à la possibilité d'observer une partie et visualiser les classements s’il ne désire pas jouer.

Lorsque qu'une partie débute, le serveur choisis aléatoirement le joueur qui commencera. A terme, le serveur enregistre les scores des différents joueurs et dresse un classement.

### Règles du jeu

Le Yahtzee se joue avec 5 dés et se finit une fois toutes les cases de la fiche de score remplies. Chaque joueur joue tout à tour et dispose de 3 lancers à chaque coup. L’objectif étant de réaliser des combinaisons qui rapportent des points. Le joueur a le choix de reprendre tous ou une partie des dés à chaque lancé, selon son gré, pour tenter d’obtenir la combinaison voulue. A chaque tour, le joueur doit obligatoirement inscrire les points qu’il a obtenu dans la fiche de score.

Le premier joueur lance les dés et choisit de les garder tous, de les rejouer tous ou rejouer seulement certains dés afin d’obtenir l’une des combinaisons.

Après le dernier jet de dé, il doit remplir la fiche de score. S’il n’obtient aucune combinaison ou n’est pas satisfait du score obtenu et pense pouvoir faire mieux, il a la possibilité d’inscrire 0 point dans la case de son choix, sachant qu’il ne pourra plus refaire cette combinaison par la suite. Le gagnant est celui qui obtient le plus de points. Pour compter les points, se référer à la fiche de score.

Lorsque le total intermédiaire est égal ou supérieur à 63 points, un bonus de 35 points supplémentaires est accordé, ce qui peut faire la différence au décompte final. Il faut donc être stratégique.

### Contraintes

* Un joueur ne peut rejoindre une partie en cours (y compris dans le cas où il s'y est déconnecté involontairement);
* Une partie ne peut être composé de plus de 6 joueurs ;
* Le protocole client-serveur n'est pas chiffré.

# Responsabilités client et serveur

## Serveur

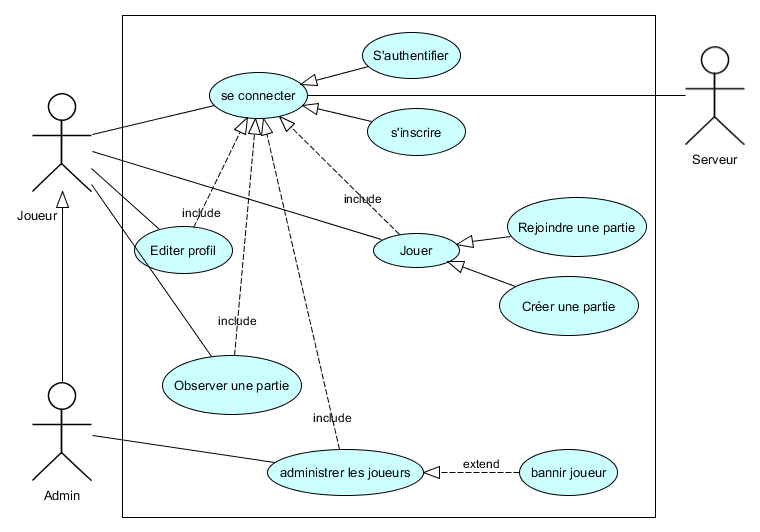
* Gestion de la base de données
* Instancier les parties
* Dresser les classements
* Mise en relation des joueurs
* Logique du jeu
* Configuration du serveur
* Valider authentifications utilisateurs

## Client

* Authentification auprès du serveur
* Mise-à-jours des vues

# Cas d'utilisation

## Diagramme



## Acteurs

Joueur : Acteur principal dans : se connecter, jouer, éditer son profil et observer une partie

Admin : Acteur principal dans administrer les joueurs

Serveur : Acteur secondaire lors de la connexion

Virer serveur du diagramme et mettre acteur secondaire pour tout, puisqu’il envoie un acknowledge pour toutes les tâches… ??

## S'inscrire sur un serveur

### Scénario principal (succès)

1. Le joueur saisi l'adresse IP du serveur, un nom d'utilisateur et un mot-de-passe
2. Le joueur confirme l'inscription
3. Le serveur envoie un « acknowledge »
4. Affichage de la confirmation du serveur

### Autres scénarios (échecs)

**Échec 1 : Le serveur est introuvable**

1. Le joueur est notifié
2. Le joueur corrige l'adresse IP

**Échec 2 : Le nom d'utilisateur existe déjà**

1. Le joueur est notifié
2. Le joueur enregistre un nom d'utilisateur différent

**Échec 3 : Le joueur annule la procédure**

1. Le joueur abandonne l'enregistrement

## S'authentifier sur un serveur

### Scénario principal (succès)

1. Le joueur spécifie le serveur auquel il veut se connecter
2. Le joueur spécifie son nom d'utilisateur et le mot-de-passe associé
3. Le joueur confirme la connexion
4. Le serveur envoie un « acknowledge »

### Autres scénarios (échecs)

**Échec 1 : Le serveur est introuvable**

1. Le joueur est notifié
2. Le joueur corrige l'adresse IP du serveur

**Échec 2 : Le serveur ne répond pas**

1. Le joueur est notifié
2. Le joueur renouvelle sa demande plus tard

**Échec 3 : Le mot-de-passe est incorrect**

1. Le joueur est notifié
2. Le joueur corrige son mot-de-passe

## Editer son profil

### Scénario Principal (succès)

1. Le joueur change ses informations
2. Le joueur confirme
3. Le serveur envoie un acknowledge

## Créer une partie

### Scénario Principal (succès)

1. Le joueur spécifie le nombre de joueurs admissible
2. La partie est créée
3. La partie est mise en attente tant qu'il n'y a pas assez de joueur
4. Le joueur rejoint la partie créée

## Rejoindre une partie

### Scénario Principal (succès)

1. Le joueur se rend sur la liste de toute les parties (voir mockup)
2. Le joueur notifie au serveur quelle partie il veut rejoindre
3. Le serveur envoie un « acknowledge »
4. Le joueur rejoint la partie

### Autres scénarios (échecs)

**Échec 1 : La partie est déjà complète**

1. Le serveur indique au joueur qu’il a rejoint la partie trop tard, et lui propose de rejoindre une autre partie ou d’en créer une nouvelle

## Observer une partie

### Scénario Principal (succès)

1. Le joueur se rend sur la liste de toute les parties (voir mockup)
2. Le joueur notifie au serveur quelle partie il veut observer
3. Le serveur envoie un « acknowledge »
4. Le joueur rejoint la partie en tant que spectateur

### Autres scénarios (variantes)

**Variante 1 : La partie a déjà commencé**

1. Le spectateur voit la partie en cours

**Variante 2 : La partie est en attente de joueur**

1. Le spectateur voit le message « En attente de X joueur(s) »

## Administrer les utilisateurs du serveur

### Scénario Principal (succès)

1. L'administrateur accomplit l'action souhaitée (Suppression, ban, ...)
2. Le serveur envoie un « acknowledge »

## Jouer

### Scénario Principal (variantes)

**Variante 1 :**

1. Le joueur lance les 5 dés
2. Le joueur indique dans quelle case il veut inscrire son score
3. Le serveur envoie un « acknowledge »
4. Le score s’inscrit dans le tableau
5. Le joueur suivant peut jouer

**Variante 2 :**

1. Le joueur lance les 5 dés
2. Le joueur relance les dés (1 à 5 dés)
3. Le joueur indique dans quelle case il veut inscrire son score
4. Le serveur envoie un « acknowledge »
5. Le score s’inscrit dans le tableau
6. Le joueur suivant peut jouer

**Variante 3 :**

1. Le joueur lance les 5 dés
2. Le joueur relance les dés (1 à 5 dés)
3. Le joueur relance les dés (1 à 5 dés)
4. Le joueur indique dans quelle case il veut inscrire son score
5. Le serveur envoie un « acknowledge »
6. Le score s’inscrit dans le tableau
7. Le joueur suivant peut jouer

### Autres scénarios (échecs)

**Échec 1 : Un joueur quitte le jeu au milieu de la partie de plus de deux joueurs**

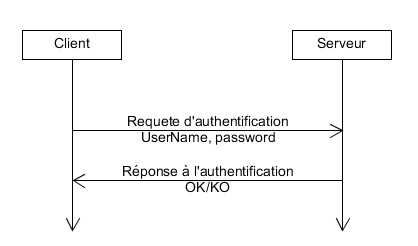
1. Les autres joueurs sont notifiés
2. Le score du joueur qui quitte est supprimé
3. La partie continue

**Échec 1 : Un joueur quitte le jeu au milieu de la partie de deux joueurs**

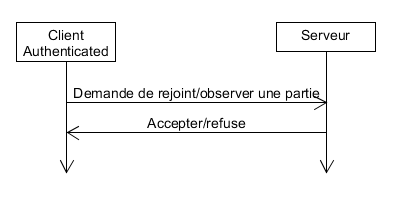
1. L’adversaire est notifié
2. Les scores des deux joueurs sont supprimés
3. La partie est annulée et supprimée

# Protocole client-serveur

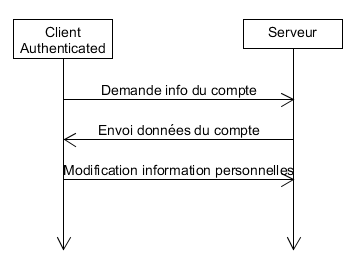
## Authentification

[](https://github.com/crabone/GEN-Yahtzee/blob/master/figures/diagrams/authentificationClient.png)

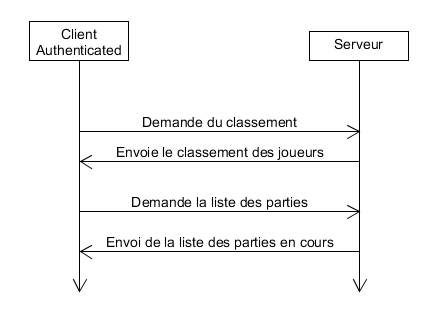
## Rejoindre/Observer une partie

[](https://github.com/crabone/GEN-Yahtzee/blob/master/figures/diagrams/DemanderObserverJoindre.png)

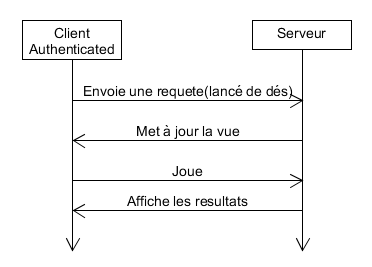
## Modification du profil

[](https://github.com/crabone/GEN-Yahtzee/blob/master/figures/diagrams/ComptePersonnel.png)

## Classement des joueurs

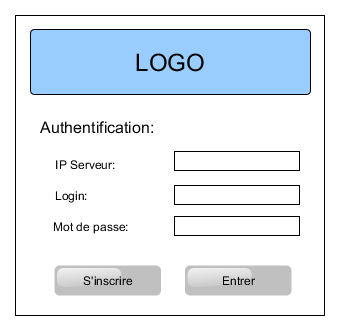
[](https://github.com/crabone/GEN-Yahtzee/blob/master/figures/diagrams/listeClassementPartie.png)

## Jouer une partie

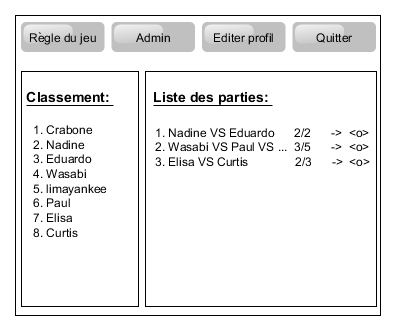
[](https://github.com/crabone/GEN-Yahtzee/blob/master/figures/diagrams/LanceDes.png)

# Ebauche UI

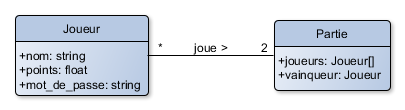
## Fenêtre d'authentification

[](https://github.com/crabone/GEN-Yahtzee/blob/master/figures/ui/Authentification.bmp)

## Fenêtre principale

[](https://github.com/crabone/GEN-Yahtzee/blob/master/figures/ui/accueil.bmp)

# Base de données

[](https://github.com/crabone/GEN-Yahtzee/blob/master/figures/BDD.bmp)

# Plan d'itération

## Itération n°1

**But**

Développer une architecture (minimale) d'une application client-serveur, permettant l'échange de messages entre plusieurs clients et un serveur

**Dates**

Du 21 avril 2016 jusqu'au 28 avril 2016

**Phase**

Implémentation et conception

**Fonctionnalités**

* Serveur pouvant gérer plusieurs clients simultanément
* Connexion d'un ou plusieurs clients sur le serveur
* Communication textuelle entre le client et le serveur
* Gestion propre des états client (Connexion, déconnexion)
* Monitoring de la part du serveur, des clients associés

**Démonstration**

* Des clients doivent pouvoir se connecter au serveur
* Visualisation des échanges entre les clients et le serveur (Par Wireshark)

**Rôles**

Fabien : Responsable normes et procédures

Kevin : Programmeur

Ibrahim : Analyste

Madolyne : Responsable documentation

Rosanne : Responsable des tests

## Itération n°2

**But**

Implémenter le protocole client-serveur. Cas d’utilisation : s’inscrire et s’authentifier sur le serveur

**Dates**

Du 28 avril 2016 au 5 mai 2016

**Phase**

Implémentation et conception

**Fonctionnalités**

* Le client et le serveur communique selon le protocole mis en place
* Le protocole contient toutes les interactions entre le client et le serveur t.q. la création d'une partie et le déroulement d'une partie
* Ce protocole sert de base pour implémenter la mécanique du jeu

**Démonstration**

* Le client et le serveur s'échange des messages selon le protocole établi (Visualisation par Wireshark)

## Itération n°3

**But**

Implémentation du moteur et de la mécanique du jeu. Cas d’utilisation : Jouer une partie

**Dates**

Du 5 mai 2016 au 12 mai 2016 (avec une réserve d'une semaine)

**Phase**

Implémentation et conception

**Fonctionnalités**

* Plusieurs clients pourront jouer des parties
* Le serveur peut accepter plusieurs parties simultanées

**Démonstration**

* Une partie (en ligne de commande) entre 3 joueurs

## Itération n°4

**But**

Implémentation de la gestion des parties et du classement. Cas d’utilisation : créer, rejoindre et observer une partie.

**Dates**

Du 12 mai 2016 au 19 mai 2016

**Phase**

Implémentation et conception

**Fonctionnalités**

* Le client peut créer une partie
* Le client peut rejoindre une partie
* Le client peut observer une partie

**Démonstration**

* Démonstrations des 3 fonctionnalités

## Itération n°5

**But**

Implémentation de l'interface graphique + cas d’utilisation : administrer les utilisateurs du serveur.

**Dates**

Du 19 mai 2016 au 2 juin 2016.

**Phase**

Implémentation et conception

**Fonctionnalités**

**Démonstration**

## Itération n°6

**But**

Dernières retouches et modification, derniers tests

**Dates**

Du 2 juin 2016 au 9 juin 2016.

**Phase**

Transition

**Fonctionnalités**

**Démonstration**