

# Projet de GEN - WarTanks

## Rapport intermédiaire

### **Auteurs :**

Armand Delessert

Simon Baehler

Benoit Zuckschwerdt

Ngueukam Djeuda Wilfried Karel

### **Destinataires :**

Eric Lefrançois

Sandy Vibert

Vendredi 24 avril 2015

## Table des matières

<b>1. Introduction</b>	<b>3</b>
<b>2. Fonctionnement général de l'application</b>	<b>3</b>
2.1. Rôle client-serveur	3
2.1.1. Le client	3
2.1.2. Le serveur	3
2.2. Déroulement d'une partie	4
<b>3. Cas d'utilisation</b>	<b>5</b>
3.1. Acteurs	5
3.2. Scénarios	6
<b>4. Développement</b>	<b>7</b>
4.1. Établissement des rôles	7
4.2. Partage des responsabilités	7
4.2.1. Programme client et interface graphique	7
4.2.2. Programme serveur	7
4.2.3. Communication réseau	7
4.2.4. Base de données	7
4.2.5. Design du jeu	7
4.3. Planning des itérations	8
4.3.1. Première itération	8
4.3.2. Deuxième itération	9
4.3.3. Troisième itération	9
4.3.4. Quatrième itération	10
4.3.5. Cinquième itération	11
4.3.6. Sixième itération	11
4.3.7. Septième itération	12
<b>5. Implémentation</b>	<b>13</b>
5.1. Serveur	13
5.2. Client	13
5.3. Client espion	13

5.4. Base de données	13
5.5. Communication client-serveur	14
5.6. Fenêtre nécessaires	14
5.7. Technologies utilisées	14

## **1. Introduction**

WarTanks est un jeu de combat de tanks en 2 dimensions. Les combats se passent sur des cartes 2D en vue de dessus. Un combat oppose 2 joueurs ou plus dans un « death match » (matche à mort), c'est donc le dernier joueur en vie qui gagne la partie.

### **Participants au projet :**

Simon Baehler, Armand Delessert, Benoit Zuckschwerdt, Ngueukam Djeuda Wilfried Karel.

## **2. Fonctionnement général de l'application**

### **2.1. Rôle client-serveur**

#### **2.1.1. Le client**

- Créer un serveur pour démarrer une partie.
- Récupère les appuis sur les touches effectués par le joueur et transmet au serveur les commandes correspondantes.
- Reçoit du serveur les mises à jour du plan de jeu et les affiche à l'écran.
- Reçoit du serveur le tableau des scores à la fin de la partie.

#### **2.1.2. Le serveur**

- Le serveur gère une partie.
- Reçoit du client les commandes du joueur (avancer, tourner, tirer, etc.).
- Met à jour le plateau de jeu.
- Retourne au client le plateau de jeu mis à jour.
- Permet la synchronisation entre les clients.
- En fin de la partie, le serveur envoie le tableau des scores aux clients et se termine.

## **2.2. Déroulement d'une partie**

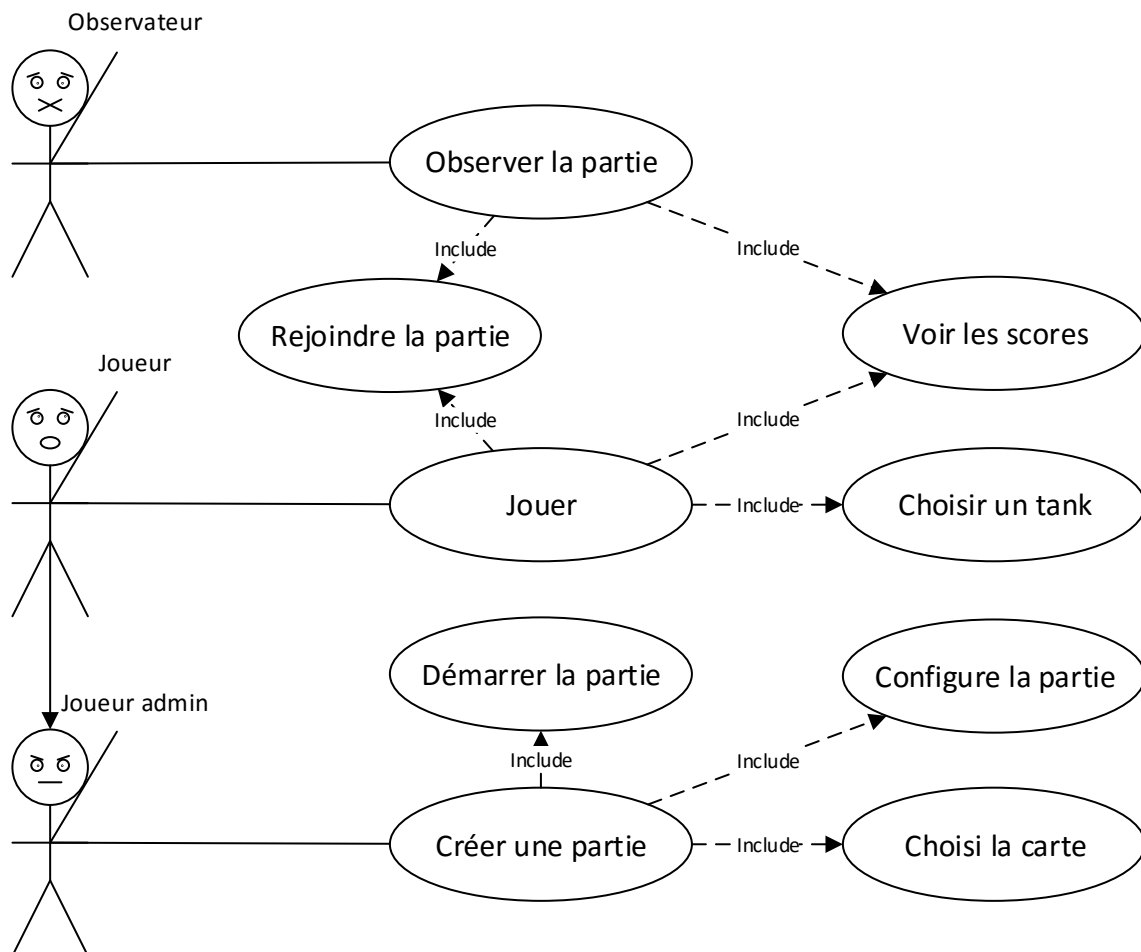
Une partie se déroule comme suit :

- Le joueur admin démarre une partie en créant un serveur.
- Le joueur admin configure le serveur (choix de paramètres tels que la carte, le temps de la partie, etc.).
- Les autres joueurs rejoignent ce serveur.
- Les joueurs (y compris l'admin) choisissent leur tank.
- L'admin démarre la partie lorsque tout le monde est prêt.
- La partie se termine dans les conditions suivantes :
  - Un joueur a gagné la partie (il est le dernier sur le terrain).
  - Le temps est écoulé.
  - L'admin a terminé la partie.
- À la fin de la partie, le tableau des scores est envoyé à chacun des joueurs.
- Le serveur s'arrête.
- Les joueurs peuvent relancer une partie en créant un nouveau serveur.

## 3. Cas d'utilisation

### 3.1. Acteurs

- Joueur admin : Il crée le serveur (nouvelle partie), s'y connecte et configure ce dernier.
- Joueur simple : Il rejoint un serveur et choisit son tank.
- Observateur : Il rejoint un serveur mais n'a aucune possibilité d'interaction avec les autres joueurs et observateurs.



## **3.2. Scénarios**

1. **Le joueur admin créer une partie :**
  - 1.1. Un serveur est créé (il gérera la partie).
  - 1.2. Le joueur admin **configure la partie**
  - 1.3. puis il **choisit une carte**.
2. **Le joueur rejoint la partie :**
  - 2.1. Il se connecte au serveur (en tant que joueur)
  - 2.2. et **choisi un tank**.
3. L'observateur **rejoint la partie :**
  - 3.1. Il se connecte au serveur (en tant qu'observateur).
4. Lorsque tous les joueurs ont rejoint la partie, **le joueur admin peut démarrer la partie**.
5. Pendant la partie, les joueurs (y compris l'admin) **jouent** chacun un tank.
  - 5.1. Les joueurs peuvent déplacer leur char et tirer sur les adversaires. Ils peuvent ramasser des bonus sur la carte.
  - 5.2. Le serveur gère la synchronisation entre les clients.
6. À la fin de la partie, **le tableau des scores est affiché** chez tous les participants.

## **4. Développement**

### **4.1. Établissement des rôles**

Chef de projet	Armand
Architecte, concepteur en chef	Armand, Wilfried
Programmeur architecture et réseau	Armand, Wilfried
Programmeur UI	Simon, Benoit
Game designer	Simon, Benoit
Graphiste	Simon

### **4.2. Partage des responsabilités**

#### **4.2.1. Programme client et interface graphique**

Simon, Benoit

#### **4.2.2. Programme serveur**

Armand, Wilfried

#### **4.2.3. Communication réseau**

Armand, Wilfried

#### **4.2.4. Base de données**

Simon

#### **4.2.5. Design du jeu**

Simon



## **4.3. Planning des itérations**

Le développement se déroulera en 7 itérations planifiées comme suit :

### **4.3.1. Première itération**

Début : Mardi 14 avril 2015

Échéance : Vendredi 24 avril 2015 13h00

Durée : 1.5 semaine

Temps de travail cumulé prévu pour cette itération : 10 heures

#### **Objectifs :**

Définition des règles et des principes du jeu, définition de l'architecture du programme, attribution des responsabilités, rédaction du rapport intermédiaire.

#### **Gestion de projet :**

- Rapport intermédiaire.

#### **Développement de fonctionnalités :**

- Un prototype de l'interface d'accueil est réalisé. La fenêtre contient les boutons suivants mais aucune action n'est lancée lorsqu'on les presse :
  - Créer une partie
  - Rejoindre une partie
  - Aide
  - Quitter

### **4.3.2. Deuxième itération**

Début : Vendredi 24 avril 2015

Échéance : Vendredi 1<sup>er</sup> mai 2015 13h00

Durée : 1 semaine

Temps de travail cumulé prévu pour cette itération : 18 heures

#### **Objectifs :**

Conception de l'architecture générale et début de l'implémentation de l'interface graphique.

#### **Développement de l'infrastructure :**

- La base de la communication client-serveur est fonctionnelle (point 2.1 du cas d'utilisation). Le client peut simplement se connecter et se déconnecter du serveur.

#### **Développement de fonctionnalités :**

- Un début d'implémentation de la fenêtre de jeu offrant la possibilité de déplacer le joueur sur la carte (point 5). Les tirs ne sont pas encore possibles.

### **4.3.3. Troisième itération**

Début : Vendredi 1<sup>er</sup> mai 2015

Échéance : Vendredi 8 mai 2015 13h00

Durée : 1 semaine

Temps de travail cumulé prévu pour cette itération : 18 heures

#### **Objectifs :**

Développement de l'architecture et de l'interface graphique sans la gestion du réseau.

#### **Développement de fonctionnalités :**

- La fenêtre de jeu implémente les fonctionnalités suivantes : la carte s'affiche ainsi que les joueurs, les déplacements du joueur sont implémentés. Les tirs sont implémentés. (point 5.1)
- Il n'y a pas encore de gestion du réseau. Les tanks adverses ne peuvent donc pas être contrôlés.

#### **4.3.4. Quatrième itération**

Début : Vendredi 8 mai 2015

Échéance : Vendredi 15 mai 2015 13h00

Durée : 1 semaine

Temps de travail cumulé prévu pour cette itération : 18 heures

##### **Objectifs :**

Début de l'implémentation de la communication client-serveur et de la synchronisation du jeu entre les clients et fin de l'implémentation de l'interface graphique du jeu.

##### **Développement de l'infrastructure :**

- La synchronisation du jeu entre les clients et le serveur (point 5.2) gère les fonctionnalités suivantes :
  - Plusieurs joueurs peuvent rejoindre la partie et jouer.
  - Les déplacements des joueurs sont synchronisés.
  - Les tirs ne sont pas encore synchronisés (les tirs adverses ne sont pas encore visibles).

##### **Développement de fonctionnalités :**

- La fenêtre de jeu est entièrement fonctionnelle (point 5.1).

### **4.3.5. Cinquième itération**

Début : Vendredi 15 mai 2015

Échéance : Vendredi 22 mai 2015 13h00

Durée : 1 semaine

Temps de travail cumulé prévu pour cette itération : 18 heures

#### **Objectifs :**

Fin de l'implémentation de la communication client-serveur et de la synchronisation du jeu entre les clients, implémentation des fenêtres des menus.

#### **Développement de l'infrastructure :**

- Synchronisation totale du jeu entre le client et le serveur (point 5.2). Plusieurs joueurs peuvent se connecter au serveur et jouer. La totalité des actions en jeu est implémentée.
- Il est possible de rejoindre une partie en tant qu'observateur (point 3).

#### **Développement de fonctionnalités :**

- Un début d'implémentation des fenêtres « Configuration du serveur » et « Configuration du joueur » (points 1.2, 1.3, 2.2).

### **4.3.6. Sixième itération**

Début : Vendredi 22 mai 2015

Échéance : Vendredi 29 mai 2015 13h00

Durée : 1 semaine

Nature de l'itération : Développement de fonctionnalités.

Temps de travail cumulé prévu pour cette itération : 18 heures

#### **Objectifs :**

Implémentation des fenêtres des menus et des scores.

#### **Développement de fonctionnalités :**

- Les fenêtres « Configuration du serveur » et « Configuration du joueur » sont implémentées (points 1.2, 1.3, 2.2). Le choix de la carte est possible lors de la configuration du serveur et le joueur peut choisir son tank au début de la partie.
- Les scores s'affichent en fin de partie (point 6).

#### **4.3.7. Septième itération**

Début : Vendredi 29 mai 2015

Échéance : Vendredi 5 juin 2015 13h00

Durée : 1 semaine

Temps de travail cumulé prévu pour cette itération : 18 heures

##### **Objectifs :**

Regroupement des fonctionnalités dans un seul projet, fin du développement du jeu et rédaction du rapport final.

##### **Gestion de projet :**

- Rapport final.

##### **Développement de fonctionnalités :**

- Le jeu est entièrement fonctionnel.

## **5. Implémentation**

### **5.1. Serveur**

- Héberge la partie.
- Gère la synchronisation des clients.
- Le serveur est créé par un des clients (celui qui créer la partie).

### **5.2. Client**

- Le premier client créer le serveur et configure la partie. Il choisit entre-autre :
  - La carte
  - Le nombre de joueurs
  - Le temps de la partie
  - Eventuellement les bonus qui peuvent apparaître
  - ...
- Les autres clients rejoignent la partie.
- Tous les joueurs choisissent leur véhicule.
- Pendant la partie, chaque client envoi les commandes (joueur qui appuie sur une touche) pour contrôler son propre véhicule.

### **5.3. Client espion**

- Spectateur de la partie. Il ne peut pas influencer le déroulement de la partie.

### **5.4. Base de données**

La base de données stocke les éléments suivants :

- Les cartes, composées des éléments suivants :
  - Un fichier contenant la carte
  - Le nom
  - La taille
  - Le nombre de joueurs
  - Une description
  - Une miniature
- Les scores, composés des éléments suivants :
  - Le score
  - Le nom du joueur
  - La carte
  - La date

## **5.5. Communication client-serveur**

Le serveur garde le plan de jeu à jour à tout moment (état de la carte, position des joueurs, état de chacun des joueurs, temps écoulé, scores, etc.). À chaque fois qu'un client envoie une commande au serveur (un joueur appuyant sur avancer par exemple), ce dernier contrôle la validité de l'action (par exemple, le joueur ne peut pas sortir de la carte) et effectue l'action sur le plan de jeu si celle-ci est valide. Le serveur envoie régulièrement le plan de jeu complet aux clients.

La communication entre les clients (clients de jeu et clients espions) et le serveur se déroulera comme suit :

- Le serveur envoie régulièrement le plan de jeu à chaque client (clients de jeu et clients espions). Ceux-ci seront ainsi capables d'afficher le plan de jeu à l'écran.
- Dans le client de jeu, lorsque l'utilisateur appuie sur une touche (pour avancer ou faire tirer le tank par exemple), la commande est envoyée au serveur.
- Le serveur vérifie la commande et l'exécute sur le plan de jeu. La prochaine mise à jour du plan de jeu qui sera envoyée par le serveur aux clients « contiendra » l'action du joueur.

## **5.6. Fenêtre nécessaires**

- Fenêtre d'accueil.
- Fenêtre de configuration de la partie (création du serveur).
  - Fenêtre de choix de la carte.
- Fenêtre de connexion au serveur (tous les joueurs).
- Fenêtre de choix du tank.
- Fenêtre de jeu. Affiche la carte et les joueurs.
- Fenêtre de présentation des scores.

## **5.7. Technologies utilisées**

Le langage de programmation : Java

Interface graphique des fenêtres : Swing

Interface graphique du jeu : Slick2D

Base de données : fichier XML.