

Projet de GEN - WarTanks

Rapport intermédiaire

Auteur :

Armand Delessert

Simon Baehler

Benoit Zuckschwerdt

Ngueukam Djeuda Wilfried Karel

Destinataires :

Eric Lefrançois

Sandy Vibert

Vendredi 24 avril 2015

Table des matières

| | |
|--------------------------------------------------------|----------|
| 1. Introduction..... | 2 |
| 2. Fonctionnement général de l'application..... | 2 |
| 2.1. Rôle client-serveur | 2 |
| 2.1.1. Le client | 2 |
| 2.1.2. Le serveur | 2 |
| 2.2. Déroulement d'une partie..... | 3 |
| 2.3. Détails techniques | 3 |
| 3. Cas d'utilisation | 4 |
| 4. Développement | 5 |
| 4.1. Établissement des rôles..... | 5 |
| 4.2. Partage des responsabilités | 5 |
| 4.2.1. Programme client et interface graphique..... | 5 |
| 4.2.2. Programme serveur..... | 5 |
| 4.2.3. Communication réseau | 5 |
| 4.2.4. Base de données | 5 |
| 4.3. Planning des itérations | 6 |
| 5. Implémentation | 8 |
| 5.1. Serveur | 8 |
| 5.2. Client | 8 |
| 5.3. Client espion..... | 8 |
| 5.4. Base de données | 8 |
| 5.5. Fenêtre nécessaires..... | 9 |
| 5.6. Technologies utilisées | 9 |

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.

2.3. Détails techniques

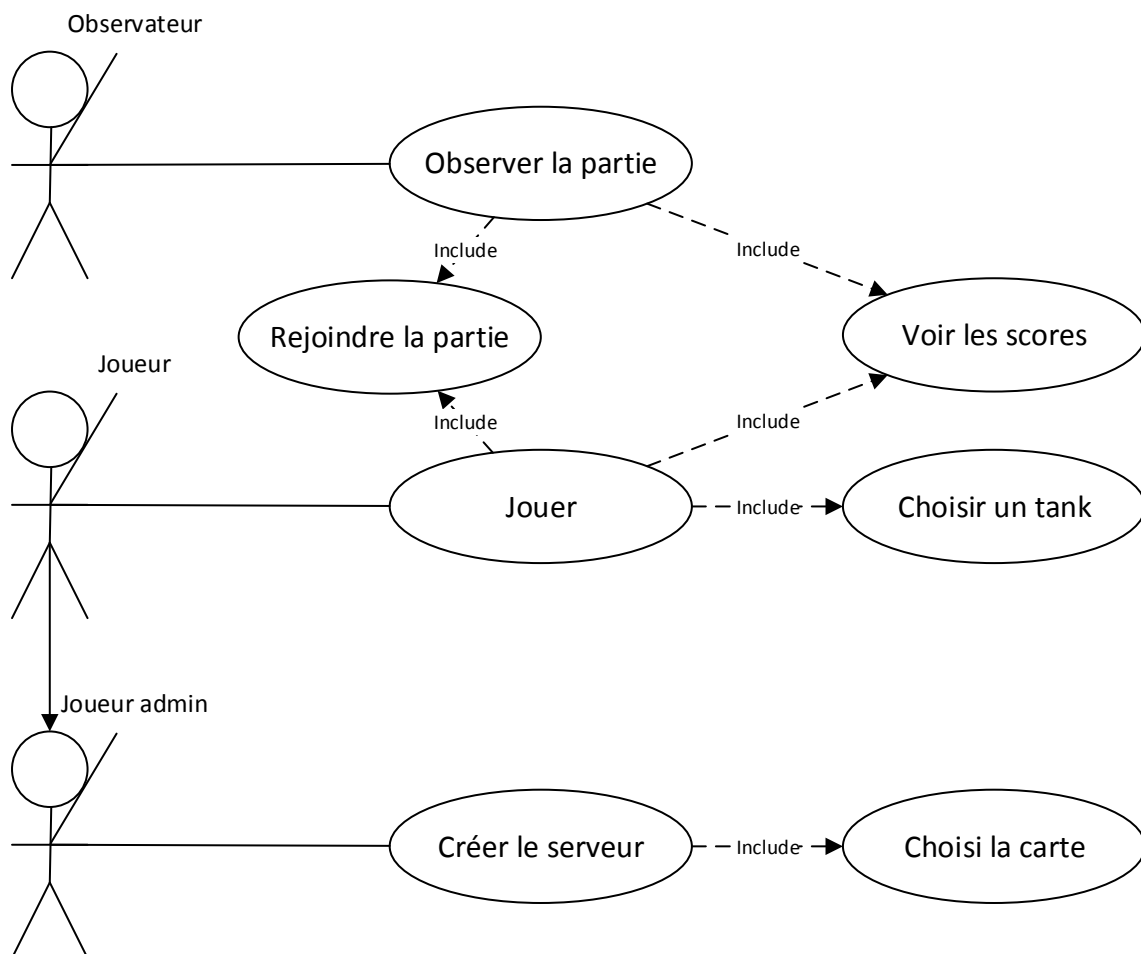
Le serveur garde le plan de jeu à jour à tout moment (état de la carte, position des joueurs, état de chacun des joueurs, temps écoulé, scores, etc.). À chaque fois qu'un client envoie une commande au serveur (un joueur appuyant sur avancer par exemple), ce dernier contrôle la validité de l'action (par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

Toutes les x ms, le serveur envoie le plan de jeu mis à jour à tous les clients.

3. Cas d'utilisation

Acteurs

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



4. Développement

4.1. Établissement des rôles

| | |
|------------------------------------|----------------|
| Chef de projet | Armand |
| Architecte, concepteur en chef | Armand, Simon |
| Programmeur architecture et réseau | Armand, Benoit |
| Programmeur UI | Simon |
| 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, Benoit, Wilfried

4.2.4. Base de données

Simon

4.3. Planning des itérations

Le développement se déroulera en 7 itérations d'une semaine chacune. Les échéances et les tâches des itérations sont planifiées comme suit :

Première itération

Échéance : Vendredi 24 avril 2015 13h00

Produits présentable :

- Un prototype de l'interface du launcher 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

Deuxième itération

Échéance : Vendredi 1^{er} mai 2015 13h00

Produits présentable :

- La base de la communication client-serveur est fonctionnelle. Le client peut simplement se connecter et se déconnecter du serveur.
- Un début d'implémentation de la fenêtre de jeu offrant la possibilité de déplacer le joueur sur la carte. Les tirs ne sont pas encore possibles.

Troisième itération

Échéance : Vendredi 8 mai 2015 13h00

Produits présentable :

- Fenêtre de jeu partiellement implémentée. La carte s'affiche ainsi que les joueurs, les déplacements du joueur sont implémentés. Les tirs sont implémentés.
- Pas encore de gestion du réseau. Les tanks adverses ne peuvent donc pas être contrôlés.

Quatrième itération

Échéance : Vendredi 15 mai 2015 13h00

Produits présentable :

- La fenêtre de jeu est entièrement fonctionnelle.
- **Synchronisation partielle** du jeu entre les clients et le serveur. Plusieurs joueurs peuvent se connecter au serveur et jouer mais toutes les fonctionnalités du jeu ne sont pas encore implémentées. Les déplacements ne sont pas forcément fonctionnels.

Cinquième itération

Échéance : Vendredi 22 mai 2015 13h00

Produits présentable :

- Synchronisation totale du jeu entre le client et le serveur. 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.
- Un début d'implémentation des fenêtres « Configuration du serveur » et « Configuration du joueur ».

Sixième itération

Échéance : Vendredi 29 mai 2015 13h00

Produits présentable :

- Les fenêtres « Configuration du serveur » et « Configuration du joueur » sont implémentées. 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.

Septième itération

Échéance : Vendredi 5 juin 2015 13h00

Produits présentable :

- Produit final.

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. 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.

5.6. Technologies utilisées

Le langage de programmation : Java

Interface graphique des fenêtres : Swing

Interface graphique du jeu : Slick2D

Base de données : SQL