

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

1. Introduction	3
2. Fonctionnement général de l'application	3
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
2.3. Déroulement détaillé	5
2.4. Les scores	6
3. Cas d'utilisation	7
3.1. Acteurs	7
3.2. Scénarios	8
3.3. Scénarios (ancienne version)	9
4. Développement	10
4.1. Établissement des rôles	10
4.2. Partage des responsabilités	10
4.2.1. Programme client et interface graphique	10
4.2.2. Programme serveur	10
4.2.3. Communication réseau	10
4.2.4. Base de données	10
4.2.5. Design du jeu	10
4.3. Planning des itérations	11
4.3.1. Première itération	11
4.3.2. Deuxième itération	12
4.3.3. Troisième itération	12
4.3.4. Quatrième itération	13
4.3.5. Cinquième itération	14
4.3.6. Sixième itération	15
4.3.7. Septième itération	16
4.3.8. Huitième itération	16

4.3.9. Présentation	16
5. Implémentation	17
5.1. Serveur	17
5.2. Client	17
5.3. Client espion	17
5.4. Base de données	17
5.5. Communication client-serveur	18
5.6. Fenêtre nécessaires	18
5.7. Technologies utilisées	18

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

L'application sera séparée en 2 parties : un client et un serveur.

Note du 28 mai 2015 : Pour assurer que le projet sera terminé dans les délais, une diminution des objectifs du projet a été décidée. Les changements apportés au projet sont visibles dans la section « Déroulement d'une partie » ci-dessous ainsi que dans le planning des itérations (itération 5 et suivantes).

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 (avancer, tirer, etc.).
- Reçoit du serveur les mises à jour du plan de jeu et les affiche à l'écran.
- Reçoit du serveur l'état du jeu (joueurs détruits, points de vie restants, etc.).
- Reçoit du serveur le tableau des scores à la fin de la partie.

2.1.2. Le serveur

- Gère la partie (une partie à la fois).
- Reçoit du client les commandes du joueur (avancer, tourner, tirer, etc.).
- Contrôle la validité des commandes et met à jour le plateau de jeu.
- Retourne aux clients le plateau de jeu mis à jour.
- Envoie aux clients l'état du jeu (joueurs détruits, temps restant, fin de la partie, etc.).
- Permet la synchronisation entre les clients.
- En fin de la partie, le serveur envoie le tableau des scores aux clients.

2.2. Déroulement d'une partie

Une partie se déroule comme suit :

- L'admin démarre une partie depuis le serveur.
- L'admin configure le serveur en choisissant les paramètres suivants :
 - ~~Choix de la carte.~~ Il n'y aura qu'une seule carte à disposition.
 - ~~Choix du nombre de joueurs.~~ Le nombre de joueurs est fixé à 2.
 - Choix du temps que dure la partie.
- Les joueurs rejoignent ce serveur (dans la limite du nombre de joueurs autorisé par l'admin).
- ~~Les joueurs choisissent leur tank et leur couleur.~~ Tous les joueurs auront le même modèle de tank. Le choix de la couleur sera automatique.
- 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 met fin à la partie.
- À la fin de la partie, le tableau des scores est envoyé à chacun des joueurs.
- L'admin peut relancer une partie ou arrêter le serveur.

2.3. Déroulement détaillé

Configuration et démarrage de la partie :

L'admin démarre une partie depuis le serveur. Une fois la partie créée, l'admin doit la configurer en choisissant la carte, ~~le nombre de joueurs~~ (le nombre de joueurs est fixé à 2) et le temps que dure la partie. Pendant qu'il configure le serveur, les joueurs rejoignent le serveur ~~et choisissent leur tank~~. Lorsqu'il n'y a plus de places libres pour les joueurs, les joueurs arrivant en plus deviennent automatiquement des observateurs de la partie. Lorsque tout le monde est prêt, l'admin lance la partie.

Déroulement de la partie :

Pendant la partie, les joueurs observateurs ne font qu'observer le déroulement de la partie sans l'influencer.

Au début de la partie, chaque joueur apparaît dans son camp sur la carte. Tous les joueurs possèdent au départ 3 vies. Les joueurs peuvent se déplacer sur les parties libres de la carte (là où il n'y a pas d'obstacle) et tirer. Les déplacements sont limités aux 4 directions nord, sud, ouest, est. Les tirs sont limités à un seul obus à la fois, ce qui veut dire que lorsqu'un joueur tir un coup il ne peut tirer un second coup seulement lorsque le premier tir aura atteint un adversaire ou un obstacle. Chaque tir reçus inflige des dégâts au tank et lorsque le tank est détruit, il réapparaît au point de départ s'il reste au moins une vie au joueur. Si le tank du joueur est détruit et que le joueur n'a plus de vie, il perd et devient observateur jusqu'à la fin de la partie. Le dernier joueur en vie gagne la partie. S'il reste plus d'un joueur en vie et que le temps de jeu est écoulé, la partie se termine.

Les bonus :

Pendant la partie, des bonus peuvent apparaître sur la carte et les joueurs peuvent les ramasser. Un bonus permet de tirer plus d'un obus à la fois.

Fin de la partie :

La partie prend fin dans les 3 cas suivants :

- Lorsqu'il ne reste plus qu'un joueur en jeu.
- Lorsque le temps de la partie est écoulé.
- Lorsque l'admin met fin à la partie.

Dans les 2 premiers cas, la fenêtre de jeu se ferme et le tableau des scores s'affiche à l'écran de chaque joueur et de chaque observateur. Dans le troisième cas, la partie s'arrête et l'admin peut choisir de relancer une partie ou de fermer le serveur.

2.4. Les scores

En fin de partie, les scores sont affichés. Le gagnant est le dernier joueur en vie ou le joueur ayant détruit le plus d'adversaires s'il reste plusieurs joueurs en vie en fin de partie.

Tableau des scores :

Voici un exemple de tableau de score tel que présenté à la fin d'un match :

		Morts				
		Joueur 1	Joueur 2	Joueur 3	Joueur 4	Total
Détruits	Joueur 1		1	1	1	3
	Joueur 2	2		2	1	5
	Joueur 3	1	0		2	3
	Joueur 4	1	2	2		5
Total		4	3	4	4	

Tableau des scores

Ce tableau résume donc le scénario où :

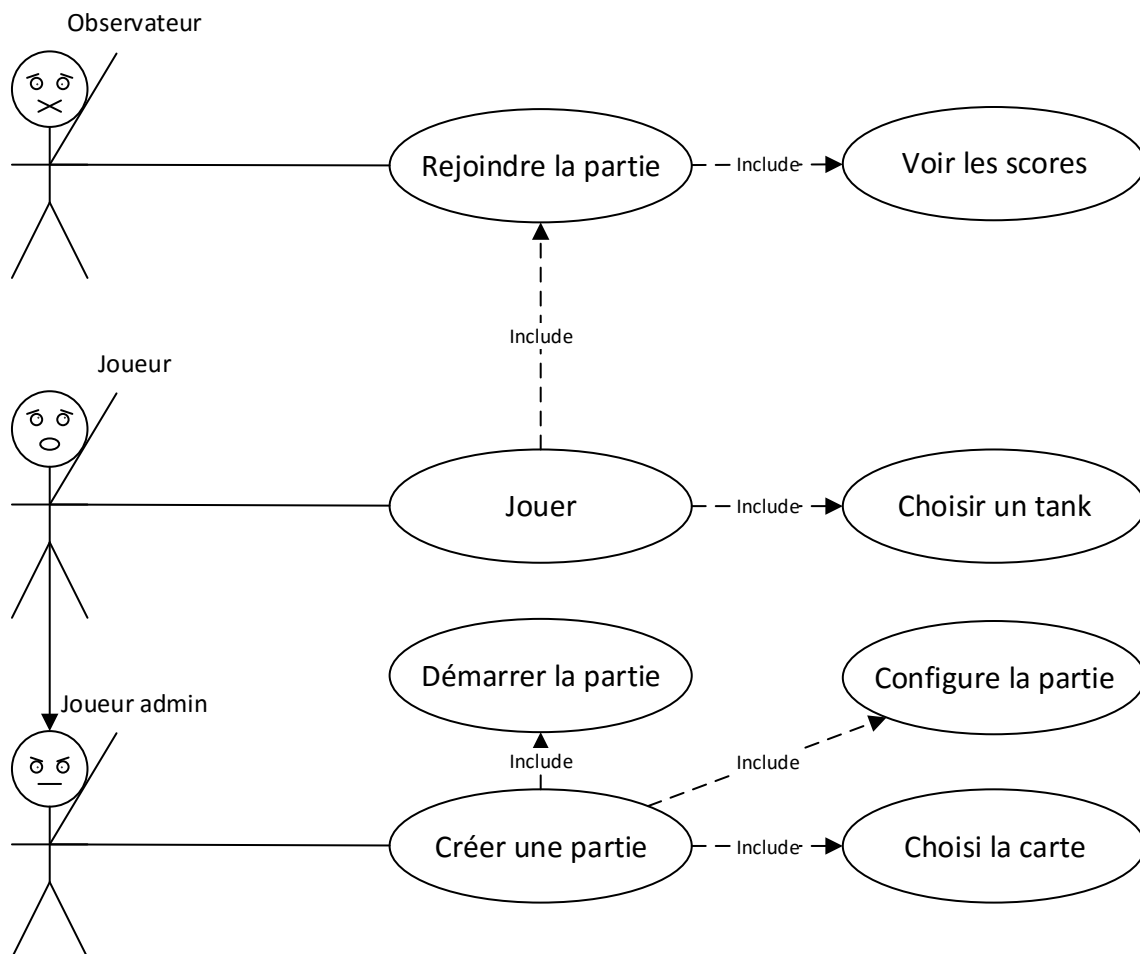
- le **joueur 1** a détruit 1 fois chaque joueur,
- le **joueur 2** a détruit 2 fois le **joueur 1**, 2 fois le **joueur 3** et 1 fois le **joueur 4**,
- le **joueur 3** a détruit 1 fois le **joueur 1**, 2 fois le **joueur 4** et n'as pas détruit le **joueur 2**,
- le **joueur 4** a détruit 1 fois le **joueur 1**, 2 fois le **joueur 2** et 2 fois le **joueur 3**.

Dans ce scénario, le dernier joueur en vie est le joueur 2. C'est donc lui le vainqueur.

3. Cas d'utilisation

3.1. 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.



3.2. Scénarios

Nouvelle version des scénarios, valable depuis l'itération 5.

1. **Le joueur admin crée une partie :**
 - 1.1. Le joueur admin crée le serveur (le serveur gérera la partie).
 - 1.2. Une fois le serveur configuré, le joueur admin rejoint les autres joueurs (~~choix du tank, etc.~~).
2. Le joueur admin **configure la partie.**
- ~~3. Le joueur admin choisit une carte.~~
4. **Le joueur joue :**
 - 4.1. Le joueur se connecte au serveur (en tant que joueur).
 - ~~4.2. Le joueur choisit un tank.~~
5. **L'observateur rejoint la partie :**
 - 5.1. L'observateur se connecte au serveur (en tant qu'observateur).
6. Lorsque tous les joueurs ont rejoint la partie, **le joueur admin peut démarrer la partie.**
7. Pendant la partie, **les joueurs** (y compris l'admin) **jouent** chacun un tank.
 - 7.1. Le joueur peut déplacer son tank.
 - 7.2. Le joueur peut tirer.
 - 7.3. Le joueur peut ramasser des bonus sur la carte.
 - 7.4. Le serveur gère la synchronisation entre les clients.
 - 7.4.1. Plusieurs joueurs peuvent rejoindre la partie et jouer.
 - 7.4.2. Les déplacements sont synchronisés entre les clients.
 - 7.4.3. Les tirs sont synchronisés entre les clients.
8. À la fin de la partie, **le tableau des scores est affiché** chez tous les participants.

3.3. Scénarios (ancienne version)

Ancienne version des scénarios, valable jusqu'à l'itération 4.

9. Le joueur admin créé une partie :

- 9.1. Le joueur admin créé le serveur (le serveur gérera la partie).
- 9.2. Le joueur admin **configure la partie**.
- 9.3. Le joueur admin **choisit une carte**.
- 9.4. Une fois le serveur configuré, le joueur admin rejoint les autres joueurs (choix du tank, etc.).

10. Le joueur joue :

- 10.1. Le joueur se connecte au serveur (en tant que joueur).
- 10.2. Le joueur **choisit un tank**.

11. L'observateur rejoint la partie :

- 11.1. L'observateur se connecte au serveur (en tant qu'observateur).

12. Lorsque tous les joueurs ont rejoint la partie, **le joueur admin peut démarrer la partie**.

13. Pendant la partie, **les joueurs** (y compris l'admin) **jouent** chacun un tank.

- 13.1. Le joueur peut déplacer son tank.
- 13.2. Le joueur peut tirer.
- 13.3. Le joueur peut ramasser des bonus sur la carte.
- 13.4. Le serveur gère la synchronisation entre les clients.
 - 13.4.1. Plusieurs joueurs peuvent rejoindre la partie et jouer.
 - 13.4.2. Les déplacements sont synchronisés entre les clients.
 - 13.4.3. Les tirs sont synchronisés entre les clients.

14. À la fin de la partie, **le tableau des scores est affiché** chez tous les participants.

À modifier : Présenter 1 scénario par bulle du graphe des cas d'utilisation. Il faut donc 8 scénarios au total.

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^{er} 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 5.1). Les tirs ne sont pas encore possibles.

4.3.3. Troisième itération

Début : Vendredi 1^{er} 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 (point 5.1). Les tirs sont implémentés (mais pas encore synchronisés) (point 5.2).
- Il n'y a pas encore de gestion du réseau. Les tanks adverses ne peuvent donc pas être contrôlés.

Reporté depuis l'itération 2 :

- Base de la communication client-serveur (point 2.1).

4.3.4. Quatrième itération

Début : Vendredi 8 mai 2015

Échéance : Vendredi 22 mai 2015 13h00

Durée : 2 semaines

Temps de travail cumulé prévu pour cette itération : 36 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 5.4) gère les fonctionnalités suivantes :
 - Plusieurs joueurs peuvent rejoindre la partie et jouer (point 5.4.1).
 - Les déplacements des joueurs sont synchronisés (point 5.4.2).
 - 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 (points 5.1, 5.2, 5.3).
- Implémentation de la gestion des collisions (point ...).

Reporté depuis l'itération 3 :

- Afficher les autres joueurs, gestion des collisions, implémentation des tirs.

4.3.5. Cinquième itération

Début : Vendredi 22 mai 2015

Échéance : Vendredi 29 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 la communication client-serveur.

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 (point 5.4.1).~~
 - ~~○ Les déplacements des joueurs sont synchronisés (point 5.4.2).~~
 - ~~○ Les tirs ne sont pas encore synchronisés (les tirs adverses ne sont pas encore visibles). (Reporté à l'itération suivante)~~
- Il est possible de rejoindre une partie en tant qu'observateur (point 5).

Développement de fonctionnalités :

- ~~• Implémentation de la fenêtre de configuration du serveur (point 1.2).~~

Réduction des objectifs du projet :

Les semaines avançant et la charge de travail à côté de GEN ne diminuant pas, nous avons décidé de réduire les objectifs du projet. Nous allons donc retirer les fonctionnalités suivantes du projet :

- Le choix du tank en début de partie : Tous les joueurs auront le même modèle de tank. La couleur sera choisie automatiquement.
- Le choix du nombre de joueurs : Le nombre de joueurs dans une partie est fixé à 2.
- Le choix de la carte : Il n'y aura qu'une seule carte disponible.

4.3.6. Sixième itération

Début : Vendredi 29 mai 2015

Échéance : Vendredi 5 juin 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 l'infrastructure :

- La synchronisation du jeu entre les clients et le serveur (point 7.4) gère les fonctionnalités suivantes :
 - Plusieurs joueurs peuvent rejoindre la partie et jouer (point 7.4.1).
 - Les déplacements des joueurs sont synchronisés (point 7.4.2).
 - Les tirs ne sont pas encore synchronisés (les tirs adverses ne sont pas encore visibles).
- Il est possible de rejoindre une partie en tant qu'observateur (point 5).

Développement de fonctionnalités :

- La fenêtre de configuration du serveur est implémentée (point 1.2). ~~Le choix de la carte est possible lors de la configuration du serveur (point 1.3) et le joueur peut choisir son tank au début de la partie (point 2.2).~~ (Objectifs retirés)
- Les scores s'affichent en fin de partie (point 6).

4.3.7. Septième itération

Début : Vendredi 5 juin 2015

Échéance : Vendredi 12 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 l'infrastructure :

- Synchronisation totale du jeu entre le client et le serveur (point 7.4). Plusieurs joueurs peuvent se connecter au serveur et jouer. La totalité des actions en jeu est implémentée.

Développement de fonctionnalités :

- Le jeu est entièrement fonctionnel.

4.3.8. Huitième itération

Début : Vendredi 12 juin 2015

Échéance : Vendredi 19 juin 2015 13h00

Objectifs :

Préparation de la présentation du projet.

4.3.9. Présentation

Date : Vendredi 19 juin 2015

Présentation du projet et rendu du rapport.

5. Implémentation

5.1. Serveur

- Héberge la partie.
- Gère la synchronisation des clients.
- Le serveur est créé par l'admin.

5.2. Client

- L'admin crée le serveur et configure la partie. Il choisit entre-autre :
 - la carte,
 - le nombre de joueurs,
 - le temps de la partie.
- Les autres clients rejoignent la partie.
- Tous les joueurs (y compris l'admin) choisissent leur véhicule.
- Pendant la partie, chaque client envoie les commandes au serveur (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 de la carte
 - La taille de la carte
 - Le nombre de joueurs pouvant jouer sur la carte
 - 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 pour permettre à ces derniers de rafraîchir l'affichage.

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.
- ~~Fenêtre de choix du tank.~~
- Fenêtre de jeu. Affiche la carte et les joueurs.
- Fenêtre de présentation des scores.

5.7. Technologies utilisées

Langage de programmation : Java

Interface graphique des fenêtres : Swing

Interface graphique du jeu : Slick2D

Base de données : fichiers XML.