

2010

ASD Tower Defense

Version 2.0 – En réseau

Rapport Intermédiaire

Aurélien Da Campo
Lazhar Farjallah
Pierre-Dominique Putallaz
Romain Poulain

GEN
HEIG-VD
07/05/2010



Table des matières

1.	Introduction.....	3
2.	Règles du jeu	4
	Les créatures	4
	Les tours	4
	Les modes de jeu	4
3.	Etapas de lancement du jeu	5
	Cas d'utilisation	5
	Acteurs principaux.....	6
	Scénario principal (succès)	6
4.	Cas d'utilisation	7
	Schéma global des acteurs	7
	Serveur d'enregistrement	8
	Acteurs principaux.....	8
	Scénario principal (succès)	9
	Serveur de Jeu	10
	Serveur Web.....	10
5.	Protocoles d'échange	11
	Serveur d'enregistrement	11
	Serveur de jeu	11
6.	Diagramme UML de classes.....	12
	Model du serveur	12
	Model du Client	13
	Réseau	14
7.	Charte graphique.....	15
8.	Rôle des participant au sein du groupe.....	17
9.	Itérations	18
	Itération 1 – Serveur d'enregistrement + Interface graphique.....	18
	Itération 2 – Serveur de Jeu + Architecture	19
	Itération 3 –Intégration du serveur de jeu + Interface du Jeu en réseau	19
	Itération 4 – Lifting de la GUI + Game Design + Amélioration Mode Solo	20
	Itération 5 – Serveur Web (facultatif)	20
10.	Conclusion	21
11.	Annexes	21

1. Introduction

Ce projet prend place durant notre 4e semestre aux seins de la Haute Ecole d'Ingénierie et de Gestion du canton de Vaud (heig-vd). Ce cours de Génie logiciel (GEN) nous propose de mettre en pratique les notions théoriques acquises en créant une application de type Client/ Serveur.

Nous avons tout de suite pensée à l'amélioration d'un jeu que nous avons créé durant nos cours d'Algorithmes et Structures de Données (ASD2) suivi durant notre 3e semestre. En effet, nous avons réalisé un jeu et ce nouveau projet est pour nous l'opportunité d'étendre ce logiciel en lui fournissant des fonctionnalités réseau.

Ce document vous présente le rapport intermédiaire du projet reposant sur une gestion de projet basé sur la méthode UP.

Pour d'éventuelles questions, nous vous fournissons le rapport de la version 1.0 du projet réalisée durant notre cours ASD2 2009-2010.

2. Règles du jeu

Le principe du jeu est de survivre à diverses créatures dont la seule capacité est d'avancer le long du chemin le plus court pour rallier leur point de départ à la zone de fin. Lorsqu'une créature atteint la zone de fin, elle fait perdre une vie au joueur. Lorsque le joueur n'a plus de vie, il a perdu.

Les créatures

Toutes les créatures prennent le chemin le plus court depuis leur emplacement jusqu'à la zone de fin. Les créatures terrestres doivent contourner les murs et les tours. Les créatures volantes peuvent survoler les tours.

Les tours

Pour se défendre, le joueur peut acheter des tours qu'il place sur son plateau. Certaines tours font des dégâts (ciblés ou de zone) d'autres permettent seulement de ralentir les créatures. Certaines ne sont efficaces que sur un certain type de créature.

Les contraintes pour la construction d'une tour est de ne pas la créer là où se trouve une créature à ce moment et il doit toujours y avoir un chemin entre la zone de départ et celle de fin.

Les modes de jeu

Le jeu, dans sa première version, était uniquement local et donc proposait uniquement un mode solo où le joueur survivait aux vagues de créatures lancées par l'ordinateur. L'intégration de fonctionnalités réseau permettent son extension à un jeu multi-joueurs.

Une première analyse nous a permis de déterminer plusieurs types de partie dont voici quelques exemples :

- **Coopération** : plusieurs joueurs s'allient contre l'intelligence artificielle et jouent sur le même plateau.
- **Coopération zone** : Idem que coopération mais chaque joueur possède une zone du plateau de jeu partagé et ne peut bâtir des tours que dans celle-ci.
- **Versus** : chaque joueur possède son propre plateau et joue seul contre tous les autres. Il gagne de l'argent périodiquement et en tuant des créatures. Pour augmenter son revenu périodique, il peut acheter des créatures qu'il envoie chez l'ennemi.
- **Domination** : un joueur est désigné pour se battre contre les autres. Le joueur seul a des caractéristiques et des bonus de meilleure qualité que les autres mais est handicapé par sa vitesse d'exécution des opérations réduite.

Nous avons retenu le mode versus uniquement car les délais ne permettent pas de les faire tous. Ce mode est inspiré d'une extension de Warcraft III (de Blizzard Entertainment) qui est la pionnière en matière de « Tower Defense ».

3. Etapes de lancement du jeu

Dans ce chapitre, nous vous présentons le cas d'utilisation du lancement de la terminaison d'une partie de jeu. Le scénario relatif est présenté à la suite de ce schéma.

Cas d'utilisation

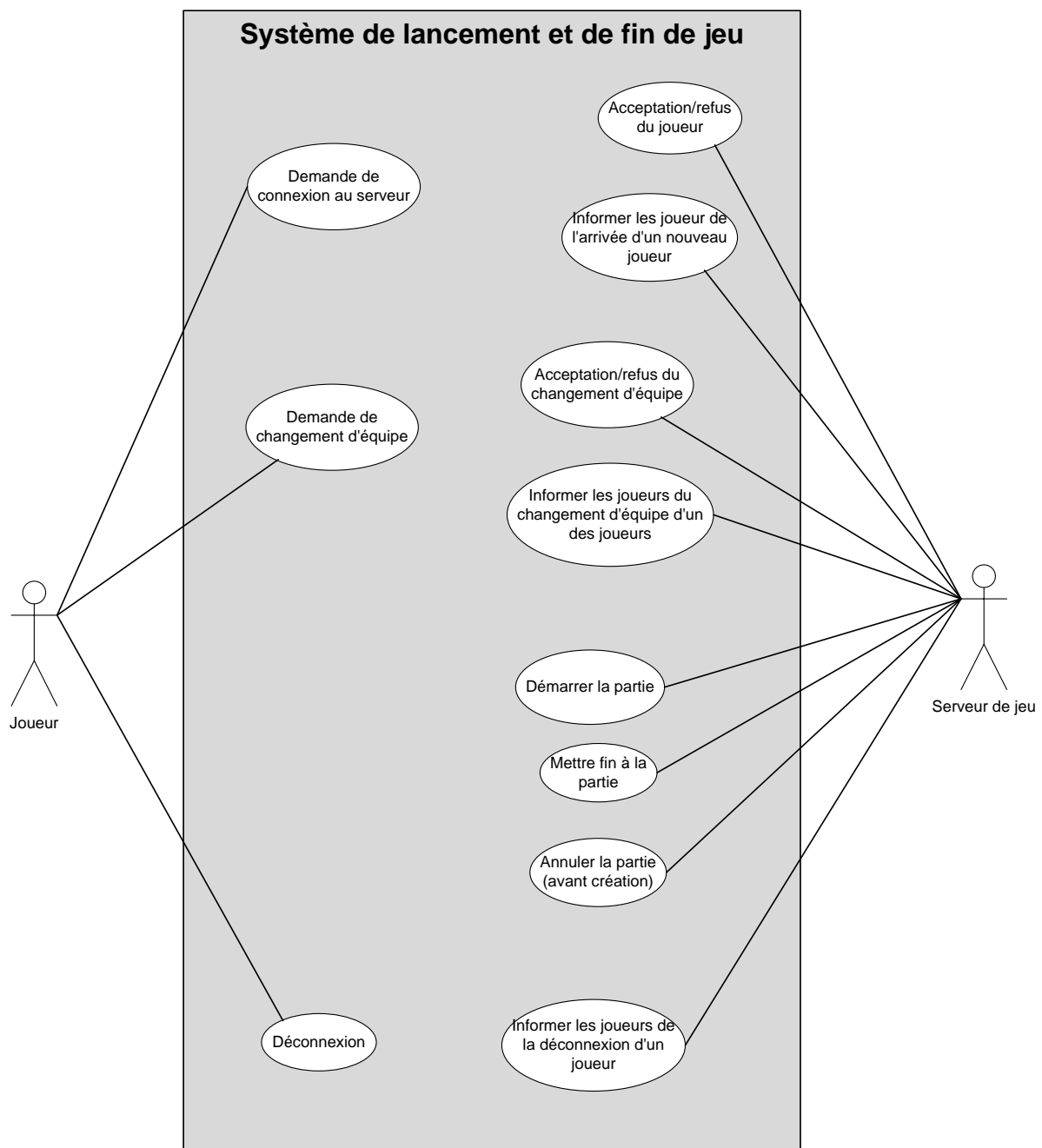


Figure 1 : Système de lancement et de fin de jeu

Acteurs principaux

- Joueur
- Serveur de jeu

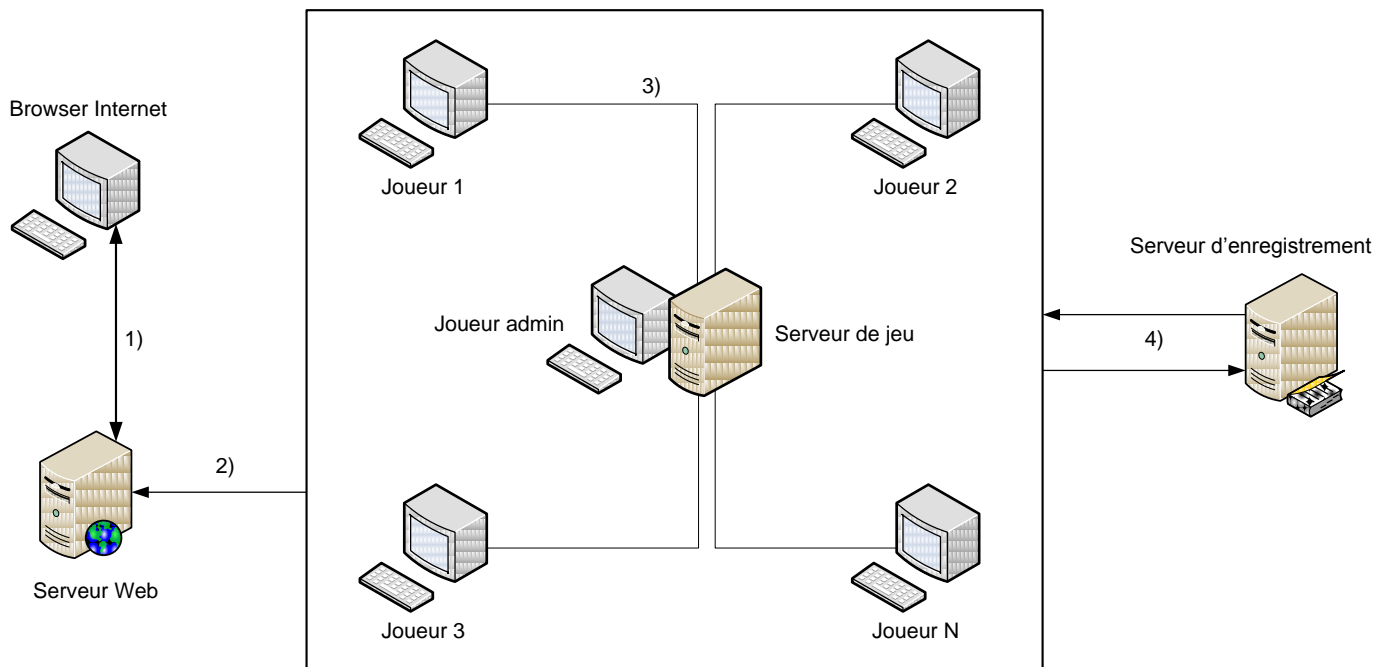
Scénario principal (succès)

Joueur	Serveur de jeu
<ol style="list-style-type: none">1. Le joueur choisit le jeu multijoueur qu'il veut créer ou rejoindre avec l'interface2. Le joueur choisit de rejoindre une partie ou en crée une3. Le joueur se connecte au serveur de jeu6. Le joueur choisit son équipe9. Le joueur joue sa partie	<ol style="list-style-type: none">4. Le serveur accepte la demande et la traite (création de la partie ou ajout du joueur à une partie)5. Le cas échéant, informe les autres joueurs de l'arrivée du dernier7. Le serveur valide le choix du joueur et informe les autres de ce changement8. Le serveur lance la partie a proprement dit10. Lorsque la fin de partie est détectée, les joueurs sont informés et déconnectés de la partie

4. Cas d'utilisation

Schéma global des acteurs

Nous vous présentons tout d'abord un schéma global mettant en œuvre les différents acteurs de l'application.



- 1) Consultation et administration des données de jeu (scores, etc.)
- 2) Envoi des données de jeu en fin de partie
- 3) Communication bidirectionnelle des données de jeu pendant la partie
- 4) Envoi et réception de la liste des parties disponibles

Figure 2 : Schéma global des acteurs

Serveur d'enregistrement

Voici le cas d'utilisation du serveur d'enregistrement avec un scénario présenté à la page suivante.

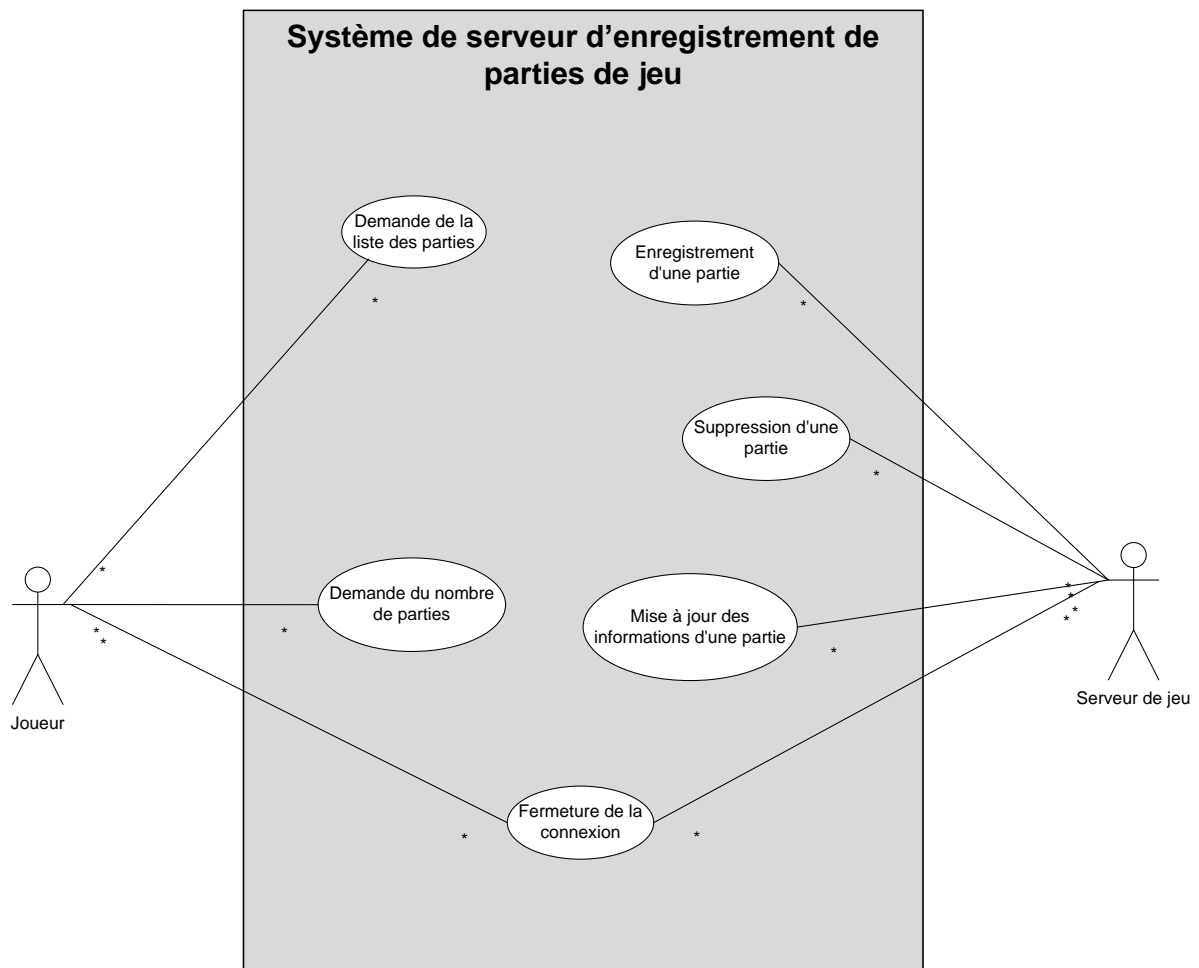


Figure 3 : Système de serveur d'enregistrement de parties de jeu

Acteurs principaux

- Joueur (d'une partie)
- Hébergeur (d'une partie)

Scénario principal (succès)

Joueur d'une partie	Hébergeur d'une partie
<ol style="list-style-type: none">1. le joueur demande la liste des parties (hébergeurs) disponibles2. Le système fournit la liste des parties (hébergeurs) en attente de joueurs3. Le joueur choisit une partie dans la liste qu'il vient de recevoir4. Le joueur se connecte à la partie5. Le joueur attend que la partie commence (que d'autres joueurs se connectent)6. La partie commence	<ol style="list-style-type: none">1. L'hébergeur crée une partie2. Le système enregistre la partie nouvellement créée dans sa base de données3. L'hébergeur attend que des joueurs rejoignent la partie qu'il vient de créer4. L'hébergeur signale au système qu'une place de moins est disponible5. L'hébergeur signale au système que la partie est complète et qu'elle va commencer6. Le système efface l'enregistrement de sa base de données

Serveur de Jeu

Voici le cas d'utilisation du serveur d'enregistrement. D'un point de vue de simplicité de réalisation et de compréhension, le scénario relatif sera présenté directement dans le protocole lié.

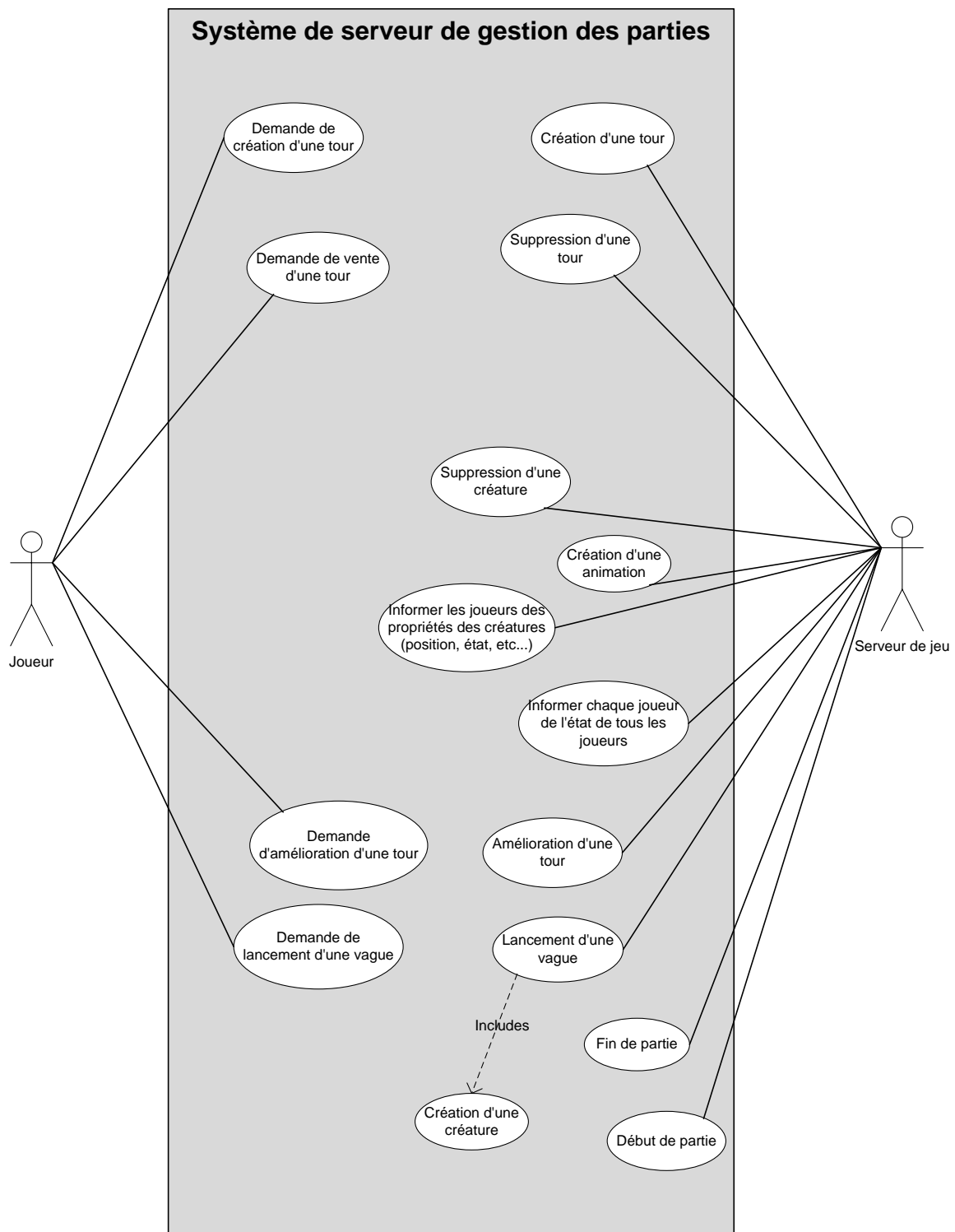


Figure 4 : Système de serveur de gestion des parties

Serveur Web

Cette étape étant optionnelle, nous n'avons pas encore réalisé les schémas relatifs au serveur web.

5. Protocoles d'échange

Serveur d'enregistrement

Le protocole du serveur d'enregistrement décrit comment communiquent les deux acteurs l'utilisant, à savoir l'hébergeur d'une partie ainsi que le joueur d'une partie. Il est principalement question de l'enregistrement d'une nouvelle partie ainsi que la récupération de la liste des parties disponibles.

Nous avons décidé d'utiliser comme format de message le JSON (*JavaScript Object Notation*), car il se prête très bien à cette situation et est très en vogue actuellement. Il s'agit principalement d'un format très standardisé que n'importe quelle entité peut comprendre, depuis n'importe quel langage.

Nous présentons en annexe la description des acteurs ainsi que le scénario de succès associés. Nous présentons de plus un schéma de communication entre les deux parties prenantes du système.

Serveur de jeu

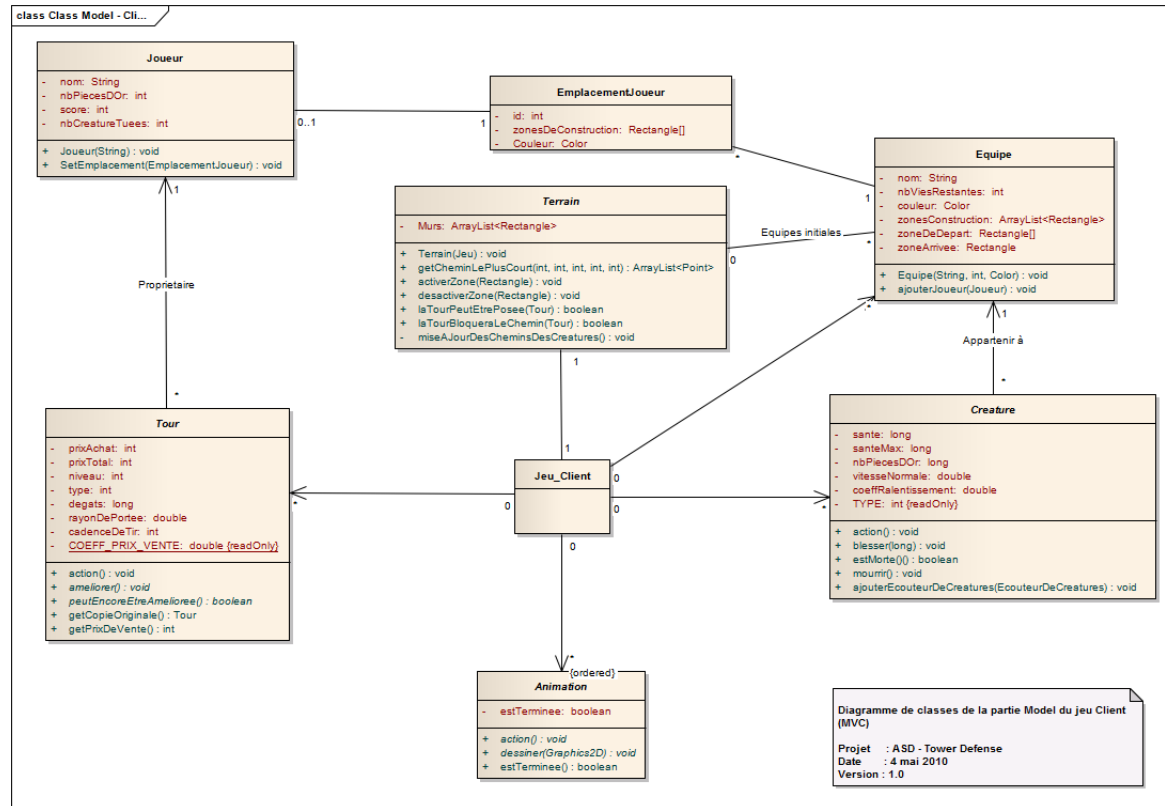
Le protocole du serveur de jeu décrit les échanges entre le client et le serveur concernant les interactions de jeu. Le client va envoyer des changements d'états ainsi que des demandes de création d'objet, demandes auxquelles le serveur va donner une confirmation en cas de réussite ou une erreur en cas d'échec.

Le serveur quant à lui va envoyer à tous les clients l'état courant de l'ensemble des objets, qu'ils aient été créés par le client cible ou d'autres clients.

Le schéma de protocole ainsi que les scénarios d'utilisations associés sont présentés en annexe « Serveur de jeu, dialogue client/serveur JSON ». Une checklist ainsi qu'un bref descriptif des services est également inclus dans le document.

Model du Client

Voici le diagramme de classes représentant la partie model (M) du client en termes d'architecture MVC.



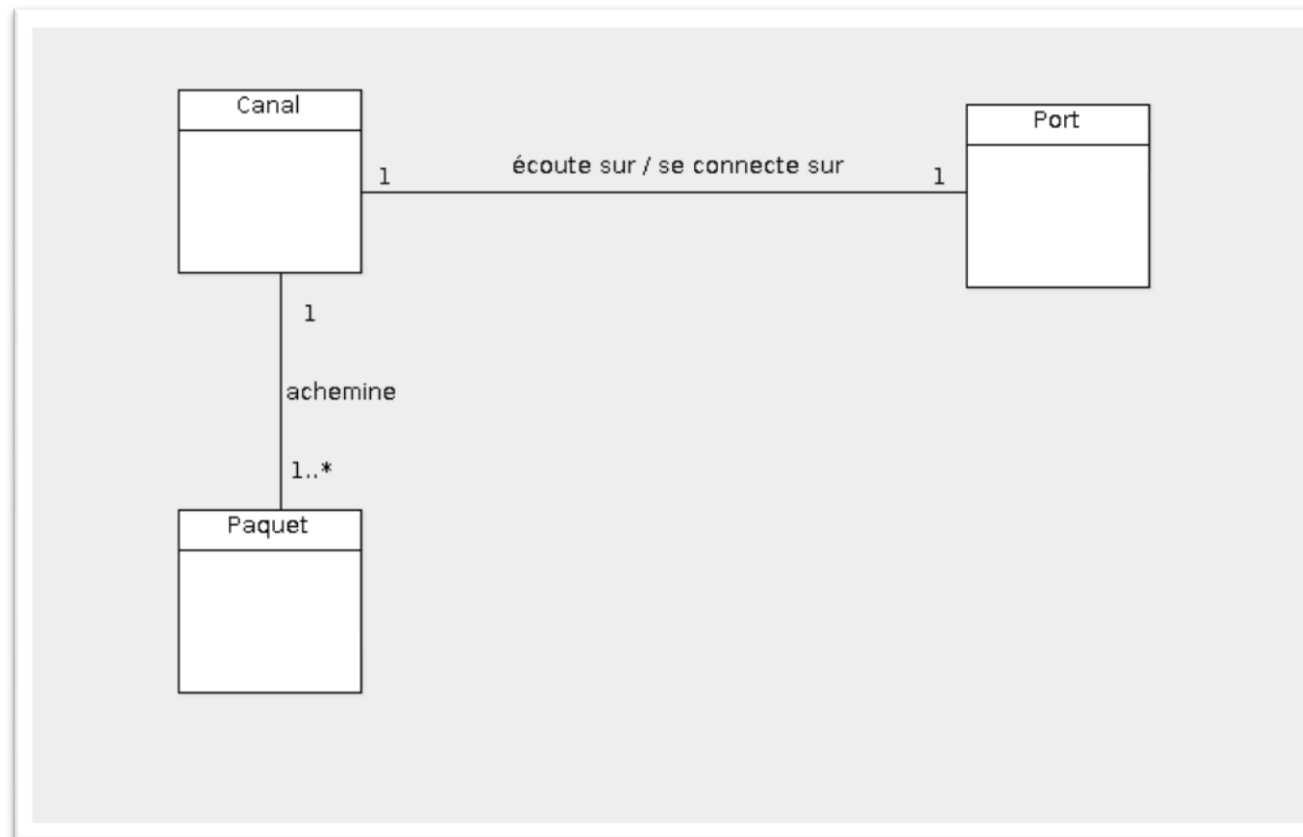
Le rôle du client est très léger, il s'occupe simplement d'afficher et de maintenir les informations reçues du serveur de jeu. De plus, les éléments qu'il possède lui permettent d'empêcher le client de faire des demandes inutiles au serveur.

Par exemple, la demande de pose d'une tour sur un mur ne sera pas envoyée au serveur mais le client empêchera directement cette action (via l'interface) grâce aux informations qu'il contient. De toute façon toutes les requêtes reçues par le serveur seront contrôlées avant de modifier le model du serveur.

Ce schéma de plus grande taille est fourni en annexe.

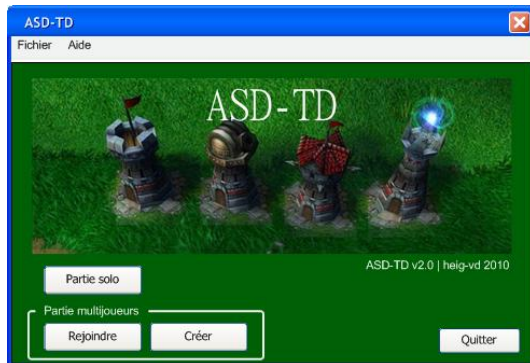
Réseau

Dans la figure suivante, nous présentons le diagramme de modélisation de domaine pour le paquetage *Reseau*, responsable de fournir les fonctionnalités réseau de base.



7. Charte graphique

Nous présentons ici nos premières idées de l'élaboration des interfaces utilisateurs de notre jeu. Nous nous concentrerons principalement sur la partie réseau.



Menu principal

Premier menu lors du lancement du jeu. Trois choix sont offerts au joueur :

1. Partie solo
2. Créer une partie multijoueurs
3. Rejoindre une partie multijoueurs

1. Mode solo

Cette fenêtre permet au joueur de faire une partie seul. 4 terrains s'offre à lui. Il en sélectionne un et la partie commence.

Note : La fenêtre de jeu solo ne sera pas présentée dans ce rapport



2. Créer une partie réseau

Le joueur introduit les caractéristiques du jeu (terrain, équipes aléatoires, etc.) et crée une partie réseau.



2.1 Attente de joueurs

La partie est créée, il faut maintenant attendre les joueurs. Le joueur qui crée la partie est considéré comme un modérateur de la partie en question. Il peut notamment changer la composition des équipes et démarrer le jeu à tous moments.





3.1 Attente de joueurs

...Le client arrive alors dans le formulaire d'attente d'autres joueurs où il peut voir les joueurs déjà connectés. Il peut changer d'équipe si de la place est disponible. Il attend ensuite que tout le monde soit connecté ou que le modérateur décide de lancer la partie.

La partie commence...

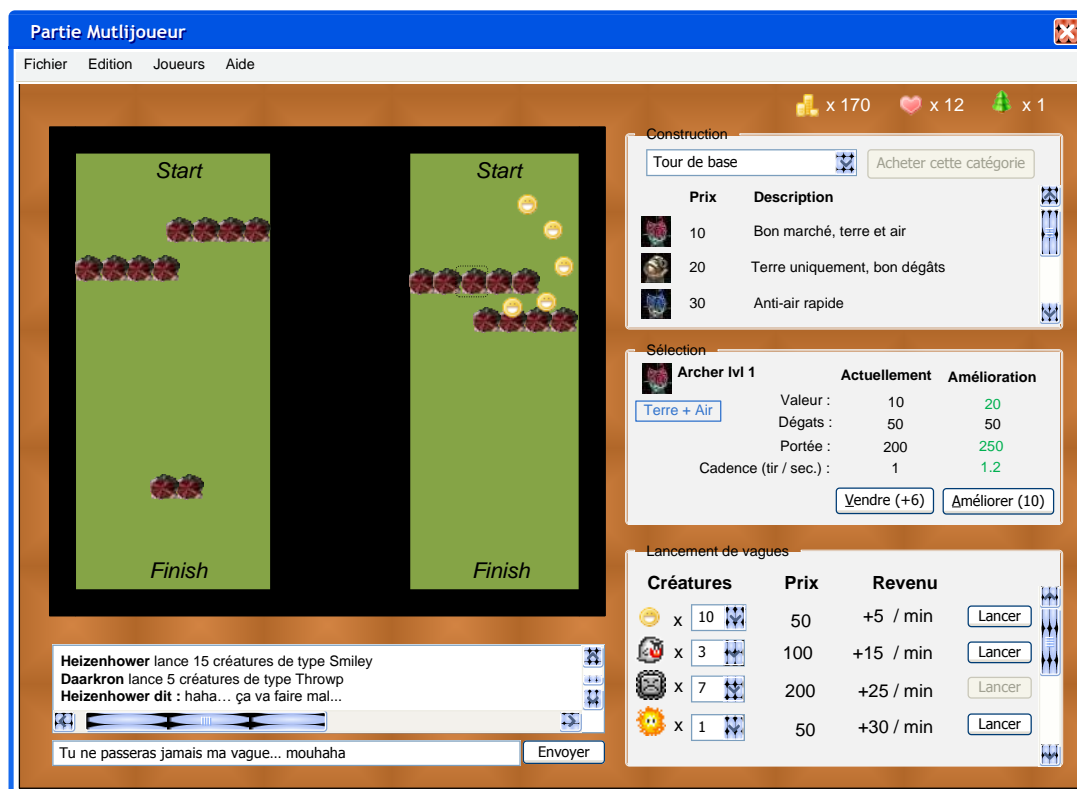
3. Rejoindre une partie réseau

Plaçons nous maintenant du côté du client qui veut se connecter. Si le serveur d'enregistrement est atteignable, le client verra la liste des serveurs disponibles avec leurs caractéristiques.

Le client sélectionne une partie et la rejoint...



4. Jeu multi-joueurs en mode Versus



Voici la fenêtre de jeu réseau. Grâce aux divers menus, le joueur pourra entre autre créer et gérer des tours, voir des informations sur les créatures (Box Sélection) et envoyer des vagues de créatures à l'équipe adverse. Un petit chat fera peut-être son apparition si le temps le permet (à priori non planifié dans les itérations).

8. Rôle des participant au sein du groupe

Voici un tableau présentant les différents rôles (standard) des membres du groupe.

	Aurélien	Lazhar	Pierre-Do.	Romain
Représentants des utilisateurs		✕		✕
Chef de projet	✕			
Analyste	✕			✕
Architecte, concepteur	✕	✕	✕	✕
Programmeur	✕	✕	✕	✕
Responsable des tests			✕	
Responsable de la configuration		✕		

9. Itérations

Dans ce chapitre nous vous présentons les différentes itérations prévues pour ce projet. Celui-ci se déroule sur **8 semaines** (phase d'initialisation comprise) à raison d'environ 4 périodes de 45 minutes par semaine. Le projet a débuté le **19 avril 2010** et il est à rendre le **11 juin 2010** avant la présentation.

Itération 1 – Serveur d'enregistrement + Interface graphique

Durée : 1 semaine – 30 avril 2010 au 7 mai 2010

Implémenter complètement la partie Serveur d'enregistrement ainsi que son interface graphique.

Le serveur d'enregistrement permet d'enregistrer les serveurs de jeu sur un serveur central afin de fournir la liste de ces serveurs aux clients. Ces derniers peuvent ensuite choisir la partie qu'ils veulent rejoindre.

Pourquoi cette itération prend-t-elle place ici ?

C'est une petite partie fournissant une bonne introduction à la notion de communication CLT-SRV notamment pour les points suivants :

- Création de notre premier protocole réseau (fixation des standards)
- Première communication CLT-SRV en Java (Création des classes de base)
- Intégration du tout dans une interface graphique cohérente. (Schéma d'interface)

Résultat attendu : le client peut enregistrer ses parties sur le serveur d'enregistrement et voir la liste de toutes les parties en attente de joueur(s) depuis une interface graphique.

Fonctionnalités attendues		
Ok	Responsable(s)	Fonctionnalité
<input type="checkbox"/>	Lazhar	Etablir une connexion client / serveur avec échange de message
<input type="checkbox"/>	Lazhar	Enregistrer une partie sur le serveur d'enregistrement.
<input type="checkbox"/>	Lazhar	Voir les parties inscrites sur le serveur d'enregistrement
<input type="checkbox"/>	Lazhar	« Désenregistrer » d'une partie sur le serveur d'enregistrement.
<input type="checkbox"/>	Lazhar	Mettre à jour les informations d'une partie.
<input type="checkbox"/>	Aurélien	Interface graphique pour l'enregistrement de la partie
<input type="checkbox"/>	Aurélien	Interface graphique pour voir les parties inscrites

Itération 2 – Serveur de Jeu + Architecture

Durée : 1 semaine - 7 mai 2010 au 14 mai 2010

Pourquoi cette itération prend-t-elle place ici ?

On prépare tous les éléments pour les fusionner ensuite (avec adaptations quasies certaines)

Création de l'application client / serveur pour le jeu. En parallèle nous commencerons la restructuration de l'architecture pour correspondre à un jeu multi-joueurs.

Il s'agit de mettre place (sans interface) une communication entre un joueur et le serveur de Jeu.

Résultat : Un protocole de communication mis en place pour l'échange de message ente le client et le serveur de jeu. Concernant la restructuration, on attend un mode 1 joueur avec exactement les mêmes fonctionnalités mais avec une architecture beaucoup plus propre.

Fonctionnalités Attendues		
Ok	Responsable(s)	Fonctionnalité
<input type="checkbox"/>	Pierre-Dominique & Romain	Implémentation de tous les messages fournis par le protocole. (liste trop exhaustive pour les citer tous, <i>se référer au protocole en annexe</i>)
<input type="checkbox"/>	Aurélien	Le mode 1 joueur fonctionne correctement et comme avant. Présentation du schéma de classe ou de domaine.

Itération 3 –Intégration du serveur de jeu + Interface du Jeu en réseau

Durée : 2 semaines - 14 mai 2010 au 28 mai 2010

Intégration du serveur dans l'architecture et le jeu fonctionne.

Le serveur de jeu devra être intégré à l'architecture de l'application (au noyau du jeu). Le client et le serveur pourront alors interagir avec le modèle (point de vue MVC) du jeu. L'interface du jeu permettra d'illustrer ces changements.

Pourquoi cette itération prend-t-elle place ici ?

Il est temps de faire fusionner tous les éléments et en faire un programme plus cohérent.

Résultat : A la fin de cette itération, le jeu doit fonctionner et tous les messages transitant entre le client et le serveur doivent être correctement traités pas l'entité réceptrice.

Fonctionnalités Attendues		
Ok	Responsable(s)	Fonctionnalité
<input type="checkbox"/>	Aurélien	L'interface permet de solliciter des actions du jeu.
<input type="checkbox"/>	Aurélien	Le client peut se connecter à une partie de jeu.
<input type="checkbox"/>	Pierre-Dominique & Romain	Le serveur et le client interprètent les messages réseau et modifie correctement le model. Ceci est visible grâce au changement du terrain de jeu.

Itération 4 – Lifting de la GUI + Game Design + Amélioration Mode Solo

Durée : 1 semaine - 28 mai 2010 au 4 juin 2010

Revoir le design et faire de notre logiciel un « vrai » jeu vidéo

Actuellement, au niveau de l'interface graphique, notre jeu ressemble plus un à logiciel applicatif qu'à un jeu vidéo. Nous aimerions dans cette itération rendre notre jeu plus attractif en créant une interface plus agréable. Il serait aussi intéressant de concevoir nos propres ressources (images / sons / etc.) car actuellement, une bonne partie de nos images sont reprises d'autres jeux.

Une bonne chose serait de revoir également toutes les valeurs liées au jeu pour le rendre plus « jouable » (*level design*). Cette partie peut paraître bénigne mais elle est cruciale et très complexe pour ce genre de jeu car il y a énormément d'éléments qui influencent la durée de vie du joueur.

Nous voulons également implémenter un système de progression dans le mode solo pour que le joueur ressente l'envie de finir complètement le jeu.

Résultat : Un programme plus esthétique, plus jouable et avec un système de progression.

Fonctionnalités Attendues		
Ok	Responsable(s)	Fonctionnalité
<input type="checkbox"/>	Lazhar	Lifting de l'interface, celle-ci ressemble plus à un jeu.
<input type="checkbox"/>	Pierre-Dominique & Romain	Adaptation des valeurs, jeu plus agréable.
<input type="checkbox"/>	Aurélien	Système de progression mise en place.

Itération 5 – Serveur Web (facultatif)

Durée : moins d'une semaine - 4 juin 2010 au 9 juin 2010

Serveur web de stockage des meilleurs scores

Il s'agit de mettre en place un serveur de web fournissant un service web de sauvegarde et récupération des meilleurs scores pour les différents terrains de jeu. Le but étant de motiver le joueur à s'améliorer (il doit avoir envie de rejouer le plus souvent possible).

Résultat : Un système permettant de sauver et voir les meilleurs scores de tous les joueurs (du monde).

Fonctionnalités Attendues		
Ok	Responsable(s)	Fonctionnalité
<input type="checkbox"/>	Romain	Mise en place du serveur web
<input type="checkbox"/>	Aurélien	Intégration du service dans l'application

10. Conclusion

Le projet avance bien, nous avons maintenant une bonne idée des choses restantes à faire et nous sommes impatients de pouvoir mettre en œuvre tous les artefacts créés dans ce rapport intermédiaire.

11. Annexes

1. Rapport de la version 1.0 réalisée durant le cours ASD2 - 2009-2010
2. Protocole du serveur d'Enregistrement
3. Protocole du serveur de Jeu
4. Diagramme de classes du Serveur
5. Diagramme de classes du client