

# Projet – Warcraft™ !

Binh-Minh Bui-Xuan

Mars-Mai 2015

Ce projet consiste à développer un modèle, simplifié, avec test embarqué du mode multi-joueur du jeu **Warcraft: Orcs and Humans**™, produit par Blizzard en 1994. Dans cet épisode de la série, comme dans tout épisode de cette série à l’innovation discutable, les joueurs dispensent des ordres à gogo à leur chers sujets – orcs ou humains – en appuyant et déplaçant furieusement leur main sur un objet rond à boutons cliquables...



## Cahier total des charges (partiel + projet)

Nous souhaitons formaliser, à l’aide du langage de spécification de services, les notions suivantes :

- les **villageois** de type **RACE.ORC** ou **RACE.HUMAN** que l’on peut déplacer dans une direction (angle à valeur entre 0 et 360), faire entrer dans des bâtiments aux effets parfois bizarres, et éventuellement faire faire des rixes dignes d’un film d’action de dernier cri,
- les **entités** composant le **terrain** de jeu :
  - les **mines** d’or dans lesquels on travaille en chantant,
  - les **hôtels de ville** où on vide ses poches au profit des esprits supérieur,
  - les **routes** sur lesquelles on se déplace deux fois plus vite,
  - les **murailles** dont l’unique utilité apparente est de se faire taper dessus dans la joie et le labeur.

## Enoncé du projet

L’objectif du projet est multiple :

1. donner une spécification complète du modèle de jeu dans le langage de spécification vu en cours,
2. réaliser à partir de la spécification une implémentation contractualisée, service et contrat sous forme de tests embarqués, en utilisant le langage de votre choix comme par exemple le Java vu en TME,
3. définir à partir de la spécification des objectifs de test pour assurer les couvertures logiques, les paires de transitions, et une suite de scénarios utilisateurs,
4. réaliser à partir des objectifs de test une implémentation des cas de tests spécifiés, en Junit par exemple,
5. réaliser deux versions complètes du jeu, dont une totalement “buggée”. Ici, peu importe si les deux réalisations sont avec interface graphique ou non. En revanche, elles doivent être en dehors de tout modèle du jeu : pas de spécification, pas de contrats, pas de tests MBT,
6. recueillir les messages d’erreur lors de la confrontation de la version buggée avec les tests spécifiés.

## Rendu du projet

Si on décide d’utiliser Java : une archive **jar** contenant

- un fichier au format PDF comprenant la spécification, les objectifs de tests, ainsi que les messages d’erreur de la version buggée du jeu,
- un sous-répertoire **src** comprenant le code Java du projet,

- un fichier de construction `build.xml` avec :
  - une cible `compile` pour la compilation du projet,
  - une cible `run` pour lancer le jeu,
  - une cible `test` pour lancer les tests sans contrat,
  - une cible `ctest` pour lancer les tests avec contrat.

Sinon, une archive `tar` contenant le PDF, le `src`, ainsi qu’un fichier `Makefile` réalisant les cibles similaires.

## Modalités d’évaluation

La note du projet reposera en priorité sur les critères suivants :

- la qualité de la spécification,
- la qualité des objectifs et des cas de tests,
- l’adéquation entre les spécifications et les contrats implémentés,
- l’adéquation entre les objectifs de tests et les tests implémentés.

## Description des services

On donne ici la liste minimum des spécifications attendues. On peut apporter des observations supplémentaires à un service particulier, ainsi que des services complets en supplément de la liste de base. D’ailleurs, la section suivante décrit quelques idées dans cette perspective.

**Attention !** Il est souvent plus intéressant de peaufiner la qualité de la spécification des services de base, plutôt que d’essayer d’avoir des services supplémentaires mal spécifiés !

### Villageois

Le service `Villageois` représente les sujets les moins favorables dans la société `Azeroth`. Ces parfaits exemples de la servitude sont distingués non seulement par leur `race` et leurs dimensions – plus précisément leur `largeur` et `hauteur` – mais aussi par leur `force`, leur `vitesse` et les précieux `points_de_vie` leur restant, que l’on suppose représenter fidèlement leur état de santé, pas celui du mental, généralement. On dispose également d’un observateur donnant leur état de service, c.à.d. si le villageois `est_mort` ou si l’on peut continuer à l’utiliser sans trop de problème. Finalement, on dispose d’un observateur indiquant la `quantité_d_or` que le villageois, à défaut de posséder, en est chargé du transport. Il est à noter que la race d’un villageois est obligatoirement `RACE.ORC` ou `RACE.HUMAN`. Lorsque la situation permet, ce service offre l’opération de `retrait` de points de vie, qui est probablement la seule chose notable que l’on peut s’attendre de tels figures.

### Terrain

La Mine d’or représente les ressources les plus convoitées dans le monde d’`Azeroth`, où les villageois travaillent dans la joie du Prophète-joueur et, généralement, aussi au profit de celui-ci. Ces objets de choix sont reconnaissables par leur dimensions (`largeur` et `hauteur`), leur quantité d’`or_restant`, et leur `etat_d_appartenance`, indiquant si oui ou non cette ressource est attribuée au joueur. Ce service offre l’opération de `retrait` de quantité d’or, et un observateur indiquant si la mine `est_laminée` : chose qui arrive dès que sa quantité d’`or_restant` n’a plus de valeur profitable.

Le service `HôtelVille` n’est pas tellement différent au service `Mine`. On peut notamment y faire un `dépôt` de quantité d’or, au lieu d’un `retrait`.

Le service `Route` permet de multiplier la vitesse des `Villageois` lorsqu’ils se déplacent dessus.

Le service `Muraille` modélise des gros tas immobiles où il n’est pas bon de se déplacer dessus. La seule fonction notable des `Muraille` est de se faire taper dessus jusqu’à démolition totale.

### Moteur du jeu

Le service `MoteurJeu` modélisera le jeu lui-même. Ce service aura comme principale opération le calcul d’un pas de jeu. Lors de ce calcul, on passera en paramètre un ordre éventuel à dispenser à au plus un villageois par pas de jeu : se déplacer selon une direction, prendre possession d’une mine abandonnée, vider ses poches au profit de l’un des joueurs, voir aller taper sur le villageois du village à côté.

Le `MoteurJeu` supervise un terrain de jeu magnifique sur lequel se trouvent un `HôtelVille` par race, quelque `Villageois`, plein de `Mine`, `Muraille` et `Route`. Le terrain de jeu est représenté par deux entités, `largeur` et `hauteur`, constamment fixées à certaines valeurs raisonnables. Les villageois sont initialement positionnés sur le terrain à moins de 51 pixels de l’hôtel de ville de leur race, leur autres caractéristiques sont initialisées

par des valeurs libres à l'imagination du lecteur. Les hôtels de ville sont par défaut attribués au joueur du même race, et contiennent 16 en quantité d'or. Une mine appartient à un joueur lorsqu'un villageois de même race en prend la possession et la visite régulièrement. Si une mine n'est pas visitée pendant 51 pas de jeu, elle devient abandonnée et son `etat_d_appartenance` change de manière correspondante. Un villageois peut visiter une mine ou un hôtel de ville s'il se trouve à moins de 51 pixels par rapport à cet objet.

Le nombre maximal de pas de jeu sera fixé à l'avance et on prendra soin de détecter la fin du jeu avec un observateur adéquat : partie gagnée ou perdue, selon le point de vue de chacun. On ne pourra entreprendre un pas de jeu supplémentaire si la partie est terminée. Les possibilités de terminaison de partie sont les suivantes : 1. l'un des joueur a réussi à amasser 1664 en quantité d'or dans son hôtel de ville mais pas l'autre, et avant que le nombre maximum de pas de jeu ait été atteint : il part (enfin) rejoindre le dîner familial dans la joie de l'accomplissement au détriment de son copain (si l'un des joueur réussit à posséder les deux HôtelVille, on suppose qu'il finira par réussir à amasser 1664 en quantité d'or, puisqu'il le vaut bien) ; 2. les joueurs, malgré eux, ne réussissent pas à s'enrichir après que le nombre maximum de pas de jeu a été atteint : la partie est finie, tristement, les jeunes gens n'ont plus qu'à se venger sur les chips.

La seule opération du service `MoteurJeu` concerne les commandes des joueurs : on mettra à jour les positions des villageois suivant ces commandes, ainsi que les profits éventuels résultant de leurs actions béates :

- visiter une mine abandonnée affecte celle-ci au joueur concerné ;
- visiter une mine implique le début de la corvée dans cette mine, en ce qui concerne le villageois ;
- une corvée dure 16 pas de jeu, pendant lesquels le villageois reste à la mine, travaillant en chantant ;
- à la fin d'une corvée de mine, le joyeux villageois réapparaît à côté de la mine en question, en soutirant 1 quantité d'or de celle-ci ;
- visiter un hôtel de ville vide les poches du villageois au profit de l'hôtel de ville, le villageois est prêt à repartir travailler dès que ses poches sont allégées.

## Suppléments

Les questions ci-dessous sont données à titre de points bonus pour le projet (l'attribution des points est variable pour chaque supplément). Les questions peuvent être traitées en partie, et dans n'importe quel ordre. Spécifier, ou apporter les modifications nécessaires à votre spécification afin de permettre les implémentations suivants.

- La sélection multiple se traduit par la possibilité de dispenser une commande (commune) à un groupe de villageois pendant un seul pas de jeu. Apporter les modifications nécessaires afin d'inclure la sélection multiple dans le *gameplay*.
- Le `GestionCombat` est un service s'occupant des rixes, parfois mortelles, entre villageois. Ajouter la possibilité de se battre entre Villageois.
- Le `GestionArtisanat` est un service s'occupant des réalisations exceptionnelles des Villageois, telle que des Route ou Muraille supplémentaire. Ajouter les modifications afin de rendre les villageois non-seulement joyeux mais aussi artisans de future Route et Muraille.
- Le `CentreNationalRechercheSpéciale` : il s'agit d'un bâtiment extrêmement coûteux à la construction, et encore plus coûteux à l'utilisation. Chaque joueur peut y dépenser (toujours...) d'énorme quantité d'or dans l'espoir de trouver (on finit forcément par trouver, toujours, quelque chose, à force de chercher, et, peut être, aussi à force de dépenser de l'or...) une amélioration dans la nature des villageois : santé, force, vitesse, etc.