

# Projet de spécialité : serveur MMORPG

## Description

Implémentation d'un serveur de jeu compatible avec le protocole du client « World of Warcraft », célèbre jeu MMORPG (« Massively multiplayer online role-playing game »).

## Equipe proposée

Floran NARENJI – ISI G1

Yann COLINA – ISI G1

Bastien ETCHEGOYEN – ISI G2

Etienne L'HER – IF G2

## Enseignant encadrant

Franck ROUSSEAU

## Taille d'équipe

4-5 élèves

## Description détaillée

Sorti en 2005, « World of Warcraft » est le titre ayant démocratisé le MMORPG. Fonctionnant sur un modèle centralisé (multiples clients – unique serveur), il utilise un protocole fermé qui lui est propre.

La finalité est de créer un serveur implémentant le protocole « World of Warcraft », de la sorte à ce que le client du jeu soit capable d'interagir (jouer) avec celui-ci.

Le but global est ici de s'intéresser aux différents aspects de la conception d'un serveur de MMORPG : réseaux, sécurité, génie logiciel, algorithmique, modélisation mathématique...

Le protocole « World of Warcraft » est ici choisi pour différentes raisons :

- Historique, car c'est le premier et le plus populaire
- Technique, car bien que très vaste, il reste relativement simple
- Pratique, car il existe une communauté active dédiée à son étude
- Industrielle, car il est façonné par des contraintes (techniques, budgétaires, politiques, légales...) bien réelles

Etant donné les moyens impliqués dans la réalisation de « World of Warcraft », il est impossible d'implémenter la totalité des fonctionnalités. Il est donc nécessaire de se restreindre à un sous-ensemble de fonctionnalités élémentaires.

Ce projet sera réalisé dans un langage récents, vraisemblablement en Go, avec une approche orientée acteur et un objectif de modularité et de « scalabilité ».

## Objectifs

- Monter en compétence dans un langage inconnu
- Monter en compétence sur un sujet inconnu et peu documenté

- Implémenter un serveur d'authentification complet et fonctionnel, soutenu par une base de données
- Implémenter des éléments d'un serveur de royaume (serveur de jeu à proprement parler) modulaires
  - o Vérification de l'authentification
  - o Création du personnage
  - o Choix du personnage
  - o Entrée en jeu
  - o Mouvements
  - o Chat
  - o Et plus, si possible.
- Implémenter une version répartie du serveur de royaume, séparant éléments de jeu et chat dans deux serveurs distincts, pilotés par un serveur maître (ou « broker »).

## Prérequis

- 1A Ensimag (algorithmique, structure de données, réseaux)
- 2A S1 Ensimag (bases de données, réseaux avancés, programmation orientée objet)

## Etat de l'art

En l'absence d'informations bibliographiques bien définie, la consultation de l'état de l'art est proposée comme alternative. A travers l'existence de World of Warcraft, de nombreuses communautés ont vu le jour. De fork en fork, il ne reste aujourd'hui plus qu'un projet actif majeur, TrinityCore.

Site : <https://www.trinitycore.org/>

Github : <https://github.com/TrinityCore/TrinityCore>

A l'heure actuelle, cet émulateur supporte la majorité du protocole, et s'efforce de rester à jour quant à la version officielle du protocole. Il est considéré relativement stable et offre une expérience de jeu convenable mais cependant nettement moins bonne que l'expérience officielle (zones d'ombres dans le protocole, manque de données, manque de contributeurs...).

## Planification

#	Intitulé	Durée	Prérequis
1	Initialisation/montée en compétence	2 sem.	
2	Eléments communs (cryptographie, serveur TCP...)	2 sem.	
3	Serveur d'authentification – prototype	1 sem.	1, 2
4	Serveur d'authentification – stable avec base de données	1 sem.	3
5	Serveur royaume – mise en place	1 sem.	4
6	Serveur royaume – gestion des personnages	2 sem.	5
7	Serveur royaume – monde vide	2 sem.	5
8	Serveur royaume – mouvement	2 sem.	6
9	Serveur royaume – chat (implémentation simple)	1 sem.	5
10	Serveur royaume – chat (implémentation répartie)	2 sem.	9