

# TP Brutes

Michel Hassenforder

## Objectifs

- Développer un protocole réseau propriétaire permettant de faire une application de type lesbrutes
- Développer une application serveur et une application cliente

## Groupes

- Quadrinôme : 2 clients et 2 serveurs dans 2 langages différents

## Délivrables

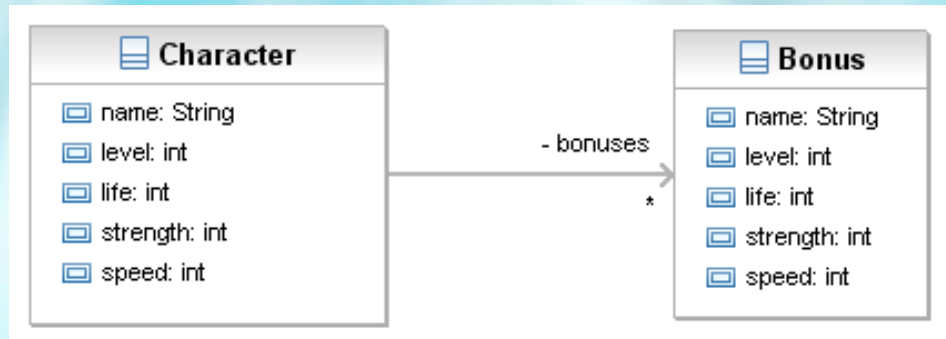
- La liste des groupes
- Une description du protocole envisagé
- Les sources du client sur une plateforme connue (pas les binaires)
- Les sources du serveur sur une plateforme connue (pas les binaires)
- Les fichiers seront déposés sur la plateforme moodle

## Plateformes connues

- Java : compatible eclipse (pas le répertoire bin)
- C++ : compatible VS (pas les répertoires release & debug)
- C# : compatible VS (pas les objects, pch, etc...)
- Php : pour le serveur essentiellement
- Autre : à négocier... vite

- Tout le projet est piloté à travers moodle
- La liste des groupes (10 décembre 2012)
  - A proposer sur la plateforme moddle avec le lien
  - <https://www.e-formation.uha.fr/moodle/mod/forum/view.php?id=8960>
- Une description du protocole envisagé (21 décembre 2012)
  - A déposer sur la plateforme moddle
- Les sources sur une plateforme connue (15 février 2009)
  - 1 fichier zip de moins de 1Mo
  - du client
  - du serveur
  - pas les binaires

- Le diagramme de classes est proche de celui-ci



## ● Comme expliqué dans le cours, il faut définir

- Stratégie
- Messages
- Sessions

## ● Stratégie

- UDP, TCP, TCPs
- mono connexion, multi connexion
- Stateless, statefull, etc...

## ● Messages

- Les messages et leurs structures physiques
- Identifiants, champs, alternatives, etc...

## ● Sessions (Assemblage de requête, réponse)

- Une requête X donne les réponses Y ou Z...
- Après requête A selon la réponse R il y a la requête B ou C...



## ● Fonctionnalités

- Le cœur de l'application est dans le serveur
- Rien n'est exposé aux clients

## ● Interface Homme Machine

- Le serveur ne possède pas d'IHM
- On peut imaginer des arguments pour le configurer mais rien de plus
- Tout se passe à travers la socket

## ● Fonctionnalités

- Le client est une IHM 'stupide'
- Aucun comportement n'est effectué localement
- Aucune ressource n'est stockée localement
  - image, son, texte, etc...
- Tous les éléments visuels seront téléchargés depuis le serveur

## ● Interface Homme Machine

- Le client est très minimal
  - Impact sur le protocole seules les description seront transférés et aucune image n'est transférée entre le serveur et le client
- Le client possède une IHM riche... très riche
  - Impact sur le protocole les description seront transférés et les images seront également transféré entre le serveur et le client

|   |     |   |       |   |       |   |         |
|---|-----|---|-------|---|-------|---|---------|
| Name  | uha | Me  | Other | Win   | Loose | state   | Connect |
| Me  |     |   |       | Other   |       |   |         |
| Name : Ensisa   |     |  |       | Name : EnscMu   |       |  |         |
| Level : 3   |     |   |       | Level : 2   |       |   |         |
| Life : 6  |     |   |       | Life : 1  |       |   |         |
| Strength : 9  |     |   |       | Strength : 3  |       |   |         |
| Speed : 7   |     |   |       | Speed : 2   |       |   |         |
| Bonuses   |     |   |       | Bonuses   |       |   |         |
|  |     |  |       |  |       |  |         |
| sword   |     | 3 bears   |       | merlin  |       | dog   |         |



- C'est juste une illustration, vous pouvez améliorer
- Les commandes sont en haut
  - Connect/Disconnect permet de se connecter au serveur si l'on utilise une socket réutilisable pour des ensembles de messages
  - State affiche alors l'état de la connexion (selon le choix de stratégie)
  - Quand on se connecte on invalide toutes les données locales (Me, Other, images, etc...) ...
  - Quand on se déconnecte on ne change pas l'affichage
  - Me, Other permet de choisir le personnage et de l'affecter à la bonne structure, un nom invalide efface la structure
  - Win, Loose permet de choisir l'issue du combat, bien entendu, il faut rafraichir les données de la fenêtre
- La structure affiche les propriétés d'une brute
  - Les propriétés de type texte sont acheminés par une requête
  - L'image viendra par une nouvelle requête.
  - L'image des bonus... même stratégie
  - Le TP n'étant pas un projet industriel, il est inutile de prévoir plus de trois emplacements pour les trois bonus maximum que nous sommes sensés avoir après les combats

- Vous disposerez d'une petite banque d'images pour composer l'IHM facilement.
- Il faudra donc étendre le modèle pour intégrer le concept d'image et de banque d'images...
- Il faudra trouver comment associer une image à un personnage...
- Les images décrivent les personnages et les bonus



Vous pouvez l'étendre ou en changer, mais cela ne compte pas dans la note





Vous pouvez l'étendre ou en changer, mais cela ne compte pas dans la note