# Organisation

Alexandre RENOUX - Pierre-Emmanuel NOVAC

24 avril 2016

## 1 Fonctionnalités

### 1.1 Mine

La quantité d'or disponible dans la mine augmente périodiquement. Ce comportement est géré côté client en Javascript à l'aide d'une méthode programmée par `window.setInterval`. On n'enverra pas une requête au serveur à chaque incrémentation pour limiter les problèmes de latence et ne pas avoir un flot de données transférées trop important. L'utilisateur pourra choisir à tout moment de transférer la quantité d'or disponible dans la mine vers son compte courant. Cette opération effectuera une requête vers le serveur avec le montant de la mine pour l'ajouter à l'or de la session de l'utilisateur et renvoyer le nouveau solde. Le client réinitialisera la quantité d'or dans la mine et mettra à jour l'affichage.

### 1.2 Guilde de mineurs

Devra être construit au préalable.
La création d'une guilde de mineurs ainsi que le recrutement de mineurs demanderont une certaine quantité d'or. Ces requêtes seront envoyées au serveur qui décidera d'honorer ou non la demande, renvoyant au client les nouvelles informations pour mise à jour de l'affichage. Un nouveau mineur sera automatiquement assigné à la mine, et dès lors il apportera un bonus sur la quantité d'or perçue.

### 1.3 Magasin

Devra être construit au préalable.
Le magasin lira la liste des objets disponibles à partir d'un fichier stocké sur le serveur (en XML par exemple). Lors du chargement de la page cette liste sera transmise au client. L'achat d'un objet se fera par l'intermédiaire du serveur, qui vérifiera le solde, débitera le compte et ajoutera l'objet à l'inventaire. Il renverra au client les informations utiles. Ces objets permettront par exemples d'augmenter la probabilité de toucher un ennemi ou augmenter le nombre de points de vie.

### 1.4 Donjon

Le joueur peut à tout moment choisir d'entrer dans le donjon. À l'intérieur du donjon, il n'aura plus la possibilité de récupérer l'or de la mine.
Les paramètres des étages du donjon et des monstres (nombre, niveau, nombre de points de vie, probabilité d'apparition, etc…) seront stockés dans un fichier sur le serveur (XML par exemple). Le déroulement du combat avec un monstre se fera côté client et l'issue sera envoyé au server. Ce dernier calculera le gain d'expérience et augmentera le niveau du joueur le cas échéant. Si il meurt avant la fin, le joueur perd de l'or. S'il réussit à battre le boss finale, il obtiendra un bonus d'or.

### 1.5 Enregistrement/chargement de partie

Toutes les données à propos du jeux sont ainsi stockées dans la session de l'utilisateur côté serveur. Il suffira de sauvegarder le contenu du tableau $_SESSION dans un fichier XML pour enregistrer la partie. Le chargement de la partie procèdera à l'opération inverse, et enverra toutes les informations utiles au client.

### 1.6 Interface utilisateur

L'interface utilisateur restera sobre : un cadre contiendra les statistiques actuelles (or, niveau du joueur, etc...), un autre cadre listera les opérations possibles énumérées ci-dessus. Enfin, une zone sera réservée à la création de bâtiment, l'achat d'objets dans le magasin et les interactions dans le donjon.

## 2 Technologie

- XHTML : pour affichage de la page web principale
- CSS : Bootstrap
- Javascript : affichage dynamique, interaction dynamique avec l'interface utilisateur
- Ajax : interaction dynamique client/serveur, envoi des données au serveur
- PHP : session, stockage des données, envoie des données au client pour affichage
- XML : sauvegarde et chargement d'une partie, paramètres du jeu
- JSON : transfert serveur/client
- Git : gestion du code
- Hébergement sur piernov.org

## 3 Planification

- 1ère semaine : mine, guilde, magasin
- 2e semaine : donjon + save/load(XML)
- 3e semaine : interface utilisateur, bugs, rapport, préparation soutenance.