Organisation

Alexandre RENOUX - Pierre-Emmanuel NOVAC

20 avril 2016

1 Fonctionalités

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 bouton envoi requête au serveur avec le montant de la mine pour l'ajouter à l'or de la session de l'utilisateur. Cette opération passera par une requête au serveur, qui mettra alors à jour le solde et qui l'enverra au client. Le client remettra la quantité d'or dans la mine à zéro.

1.2 Guilde de mineurs

La création d'une guilde de mineurs demandera une certainer quantité d'or. Puis le recrutement de mineurs requierera aussi une 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

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. bouton → réclamation d'un certain objet au serveur parmi une liste prédéfinie → renvoi l'objet au clietn pour affichage Les objets incluence lde pouvoir d'attaque (pourcentage de chance de toucher)

1.4 Doinjon

→ bouton → entre dans le donjon et lance le combat - déroulement du combat côté client (perte de point de vie et victoire/défaite) ensuite envoi du résultat du combat côté serveur : tableau des différents monstres à combattre dans l'ordre À l'issue du combat seulement le résultat est envoyé au serveur car on récupère le niveau du monstre et le nombre d'expéreince gagné associé dans le tableau incrémentation et augmentation de niveau potentiele "n" monstre par étage Si le boss est vaincu, appelle fonction de fin du donjon

1.5 bouton enregistrement/chargement de partie

: normalement toutes les données intéressantes sont dans $SESSION té serveur on transforme le tableau en fichier XML bouton load : opération inverse càd on remplkit $SESSION avec les données du XML et on envoi tout au client

1.6 Interface utilisateur

2 Technologie

XHTML : pour affichage de la page web principale CSS : Bootstrap style des boutons 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 : pour sauvegarder et charger une partie JSON : structure syntaxique pour 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.