

Projet de BDD : Gestion au Moyen Age

PHÉLIPOT PASCAL

NORO GEOFFREY

RALIJAONA TIONA

RIMOUX QUENTIN

7 décembre 2015

Table des matières

1	Introduction	1
1.1	L'idée	1
1.2	Répartition des rôles	1
1.3	Mise en œuvre	1
1.4	Gestion de projet	1
2	Représentation des données	2
2.1	Modèle-Vue-Contrôleur	2
2.2	Base de Données	2
3	Gestion des tours	3
3.1	Les possibilités	3
3.1.1	Ajax	3
3.1.2	Tache cron	3
3.2	Mise en œuvre	3
4	Possibilités d'améliorations	4
5	Conclusion	5
5.1	Avis personnels	5
5.1.1	Noro Geoffrey	5
5.1.2	Phélipot Pascal	5
5.1.3	Ralijaona Tiona	5
5.1.4	Rimoux Quentin	5
5.2	Avis Global	5
6	Annexes	I
6.1	Base de données	I
6.1.1	MySQL prévu	I
6.1.2	MySQL final	I

1 Introduction

1.1 L'idée

Le projet est un jeu par navigateur orienté stratégie en multijoueur. Chaque joueur dispose d'une base sur une carte. Il a la possibilité d'y construire des bâtiments, de gérer ses ressources et de lancer la recherche de nouvelles technologies. Les ressources sont générées chaque minute pour chaque utilisateur et il faut améliorer ses bâtiments pour en produire plus. Le but du jeu est d'attaquer les autres joueurs afin de gagner des ressources et d'améliorer sa base au maximum.

1.2 Répartition des rôles

1.3 Mise en œuvre

- Carte du monde La carte a été faite par Quentin.
- Moteur du jeu
La création du moteur du jeu a été attribuée à Pascal. Il doit réaliser la gestion des tours.
- Affichage Geoffrey et Tiona ont été chargés de l'affichage et la gestion des différents modèles.
- Données
Tout le monde a participé à l'ajout de nouvelles unités et de bâtiments.

1.4 Gestion de projet

Pour gérer ce projet nous avons bossé tout ensemble simultanément grâce à la plate-forme Github, qui permet de gérer la version de son code et de fusionner les travaux de chacun.

Le projet est accessible en ligne à l'adresse : <http://www.github.com/pasterp/GestionMiddleAge>

2 Représentation des données

2.1 Modèle-Vue-Contrôleur

Pour concevoir notre projet, nous avons souhaité nous orienter vers une organisation de notre code en MVC (Modèle-Vue-Contrôleur) avec du PHP orienté objet. Ce système nous a permis de bien structurer notre code. Nous avons donc créé 3 répertoires permettant de ranger les fichiers modèles, vues et contrôleurs de chacune des tables principales. En ce qui concerne les tables de jonctions quand à elle ne sont présentes uniquement dans des fonctions introduites dans les modèles des autres tables.

2.2 Base de Données

(Voir Annexes Page I) La base de données contient les joueurs, les cases de la carte,

3 Gestion des tours

Pour gérer les actions des joueurs, on a besoin de la notion du temps. Il faut donc que notre site opère certaines opérations au fil du temps. Il va devoir attendre une certaine durée (qu'on appellera cycle) qui a été définie à 15 minutes et ensuite lancer des actions telles que la génération des ressources, l'avancée d'une armée, etc...

3.1 Les possibilités

3.1.1 Ajax

L'AJAX (Asynchronous JavaScript and XML) est un outil permettant via un Javascript de modifier la structure d'une page en direct. Ce dernier peut donc appeler les fonctions désirées quand un utilisateur est présent sur une page.

L'avantage de cette méthode est qu'elle est transparente à l'utilisateur et qu'elle permet de faire des modifications dynamiques sur la page qu'il a devant lui.

L'inconvénient est qu'il est nécessaire d'inclure un fichier supplémentaire et que les appels seront très concentrés sur le serveur lors de l'utilisation par un grand nombre d'utilisateur.

3.1.2 Tache cron

La tâche Cron est un système au niveau système qui permet de lancer un script à des dates et heures précises. Néanmoins Cette fonctionnalité demande des droits spécifiques au système et ne peut donc être mise en oeuvre à l'IUT. Cette méthode aurait perdu de faire un appel minimum aux fonctions (juste aux horaires désirés) ;

3.2 Mise en œuvre

La solution retenue est hybride : elle est codée en PHP mais est appelée par l'utilisateur.

Chaque page chargée entraîne une vérification de l'heure du dernier cycle, PHP compare ensuite si ce cycle est assez ancien pour faire un nouveau cycle (900 secondes dans notre code) et si c'est le cas il lance les fonctions de génération de ressources.

4 Possibilités d'améliorations

5 Conclusion

5.1 Avis personnels

5.1.1 Noro Geoffrey

Lors de ce projet nous avons beaucoup travailler en équipe malgré une gestion assez mauvaise du projet qu'il faudra certainement que l'on revoie pour nos prochain projet en commun, mais l'utilisation de github nous a facilité sa réalisation car cela nous permettait de toujours avoir une version à jour du projet et de pouvoir mettre en commun chacune de nos avancés lors de nos travaux individuel. Je pense que la chose la plus importante à revoir est donc notre gestion du temps.

5.1.2 Phélipot Pascal

5.1.3 Ralijaona Tiona

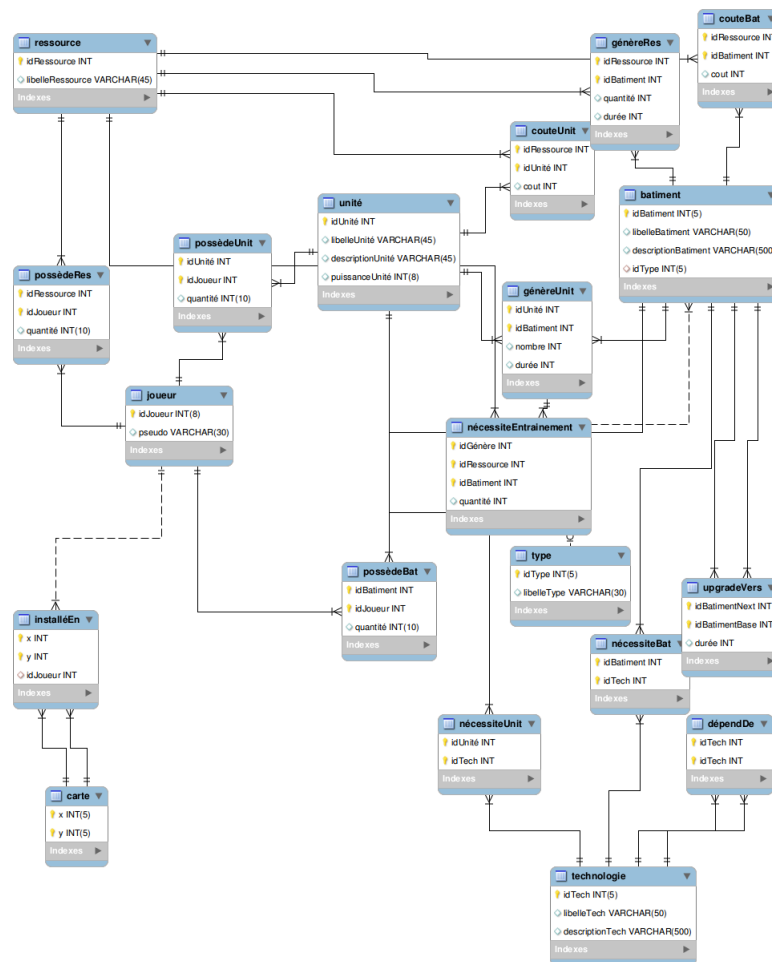
5.1.4 Rimoux Quentin

5.2 Avis Global

6 Annexes

6.1 Base de données

6.1.1 MySQL prévu



6.1.2 MySQL final

