

DROP DA BOMB

LOIGNON Lucas



TABLE DES MATIERES

Création de la base de données	3
Création des objets de View	3
Servlet de Boutique.....	3
Servlet de Compte.....	4
Création des cartes	4
Pages JSP	4
Création de comptes.....	4
Visualisation de la Boutique	5
Feuilles de style (CSS).....	5
Supplément	5
Création d'un serveur de jeu	5
Création d'un client de jeu	6
Création d'une interface de jeu	6
Design et création de cartes jouables.....	6

CREATION DE LA BASE DE DONNEES

Cette tâche a été réalisé par l'ensemble du groupe. Nous avons commencé par une étude des besoins « immédiats », suivi par une étude par anticipation des besoins futurs.

En effet, puisque nous travaillons sur ce projet avec des objectifs au-delà du cadre du projet « scolaire », c'est-à-dire avec la vraie ambition de produire le jeu et le rendre public, nous devons construire une base de données la plus complète possible, fiable et robuste. Le but est de réfléchir suffisamment lors de la conception pour avoir à effectuer un nombre minimal de modifications dans le futur.

Nous avons donc construit un MCD complet, ouvert à l'extension et optimisé (redondance minimale, efficacité des tables/relations).

CREATION DES OBJETS DE VIEW

Les Objets de View sont des Objets Java (des fichiers Java écrits sous forme « orienté objet ») qui représentent les différentes structures « utiles » de notre base de données.

Ce que j'appelle « structures utiles » dans cette section sont les différents objets, les différentes tables qui matérialisent des composants concrets de notre jeu/site, tels qu'une Carte de jeu, une Icone de Carte, un Deck de cartes, etc...

Ces objets ont été créés afin d'être manipulés facilement et efficacement par les Servlets ; il se trouvent dans le dossier « View » (lui-même contenu dans le dossier « src ») de notre architecture.

NB : Un objet dans notre cas est un fichier qui comprend : les attributs de notre objet, les accesseurs et modificateurs (« setter/getter » des attributs et les constructeurs (construit l'objet en remplissant ses attributs par les valeurs fournies).

SERVLET DE BOUTIQUE

J'ai réalisé une des fonctionnalités de la partie « Boutique » de notre site : la partie Affichage.

Il s'agit du fichier « ShopViewServlet.java » ([/src/ShopServlets/](#)).

Ce fichier traite la requête du client désirant voir la boutique, en lui renvoyant les informations à afficher (ces informations sont alors traitées pour afficher à l'écran par la page Jsp coté client).

Les informations à afficher sont envoyées au format JSON et contiennent, entre autres, les packs que le client peut acheter (disponibles en boutiques et pas encore achetés), les boosts que le client peut acheter (« »), les icones que le client peut acheter (« »), etc...

SERVLET DE COMPTE

J'ai réalisé une des fonctionnalités de la partie « Création de Compte » de notre site.

Il s'agit du fichier « CreateAccountServletNavigator.java » ([/src/AccountServlets](#)).

Cette Servlet agit comme un intermédiaire entre la page coté client et le fichier en charge des opérations relatives au Compte coté serveur. Elle reçoit donc un formulaire complété par le client, le fait « vérifier » et traiter par le fichier AccountManager.java (voir la section [Création de Comptes](#)) et s'occupe de rediriger le client en fonction du résultat de l'opération (Réussite, échec pseudo utilisé, échec email utilisé, etc..).

CREATION DES CARTES

J'ai imaginé, dessiné et conçu les visuels des différentes Cartes du jeu.

Réalisation sous Photoshop.

PAGES JSP

CREATION DE COMPTES

J'ai participé à l'aspect Gestion des comptes en m'occupant spécifiquement de la partie Création de compte.

Cette partie concerne donc les fichiers [/web/account.jsp](#) et [/src/Manager/AccountManager.java](#).

Réalisation de la page coté client (.jsp) en charge d'afficher l'interface 3-step « nouvel utilisateur » (formulaire de création de compte, lien pour le téléchargement de la future application, et lien pour la connexion au compte créé).

Conformément à notre architecture et au fonctionnement MVC, entre la page coté client et le traitement purement 'serveur', se trouve la Servlet (voir la section [Servlet de Compte](#) de ce rapport) qui agit comme un intermédiaire.

Notre fichier AccountManager.java va donc procéder aux diverses vérifications et aux diverses requêtes en base de données, à partir du formulaire transmis par la Servlet. Suite à ça, notre fichier va renvoyer à la Servlet le résultat de l'opération de création de compte qui va à son tour rediriger ou renvoyer au client l'action adaptée au résultat.

VISUALISATION DE LA BOUTIQUE

J'ai réalisé la page (.jsp) de visualisation de la boutique. Cette page affiche au joueur connecté les items disponibles (packs, boosts, icones, skin cartes, skin maps) en fonction de leurs disponibilités et du nombre d'achat maximum autorisé pour un compte.

Elle « contacte » (via la méthode POST) la Servlet en charge de cette fonctionnalité (voir la section « Servlet de Boutique » de ce rapport), qui lui renvoi au format JSON les informations à afficher. Ces informations sont reçues et traitées via le Javascript associé à la page JSP ; ce dernier parse le JSON et affiche dans chaque section (les classes d'items) les items disponibles pour le client.

Fichiers concernés : </web/WEB-INF/boutique.jsp> et </web/js/boutique.js>. Voir également la section [Servlet de boutique](#).

FEUILLES DE STYLE (CSS)

Création d'une « charte graphique » pour l'ensemble du site :

- Choix des couleurs principales
- Choix du Layout général (barre de navigation, position de l'icône principale, etc..)
- Choix du background des pages du site
- ...

Création de feuilles de style (.css) pour chaque page.

Analyse et choix graphiques afin de « styliser » au mieux les éléments spécifiques de chaque page (Boutique, Administration, Gestion des Decks, Création de Compte, Gestion de Compte).

SUPPLEMENT

Cette section concerne le développement du jeu pour clients web.

Un rapport détaillé sera fourni par la suite pour détailler les différents aspects de ce développement.

CREATION D'UN SERVEUR DE JEU

Réalisation d'un serveur de jeu en Node.js ; les fonctionnalités principales sont les suivantes : gestion du « matchmaking » (gérer les connexions et créer des parties de deux joueurs) et gestion du match (tours des joueurs, cartes posables et cartes posées, effets en cours, barre de « vie », etc...)

CREATION D'UN CLIENT DE JEU

Réalisation de la partie « back-end » du jeu coté client. Il s'agit notamment de traitements qui agissent sur l'interface de jeu en fonction des différentes informations communiquées par le serveur de jeu.

CREATION D'UNE INTERFACE DE JEU

Réalisation d'une interface web responsive (adaptée à tous médias et toutes tailles d'écran) dynamique de jeu en HTML/CSS/Javascript.

Choix du Layout, du placement des éléments, des couleurs, etc... afin de proposer une interface simple, moderne, et « fun » (qui donne envie de jouer, qui suggère un gameplay amusant).

DESIGN ET CREATION DE CARTES JOUABLES

Création de cartes de jeu en vue de créer un ensemble de cartes qui permettent de proposer un gameplay amusant, complexe et équilibré. Choix du type de carte, des effets ou non, des couts en « poudre » (une valeur dynamique permettant de poser les cartes), etc... et création d'un visuel unique et explicite pour chaque carte.