

Nom du projet : « Drop Da Bomb »

Groupe du projet : VINCENT Pierre, DEUTSCH Rémi, LOIGNON Lucas

Introduction

Principe du jeu : Drop Da Bomb est un jeu de plateau/cartes pour smartphone (Android).

*Il se base sur un ensemble de cartes (**Les Cartes de Jeu**) collectionnables (**Inventaire**) par un joueur (**Compte**) afin de composer une ou plusieurs « main de jeu » (**Deck**). Ces cartes peuvent être obtenues de diverses manières mais principalement via la boutique (**Le Shop**) qui permet d'acheter avec une certaine monnaie des lots de cartes (**Packs**). Ils existent différents types et raretés pour les cartes.*

*Une partie (**Match**) oppose deux joueurs, dans une arène spécifique avec des règles et des modes de jeu précis ; à tour de rôle les joueurs peuvent utiliser une carte de leur Deck pour tenter de faire basculer l'objectif (« la bombe ») dans le camp de leur adversaire.*

Si la bombe explose ou se trouve dans le camp d'un joueur à la fin du temps imparti, la partie est perdue par celui-ci ; elle est gagnée par l'autre qui pourra alors remporter des récompenses à déterminer.

Nous allons réaliser l'ensemble des fonctionnalités décrites brièvement ci-dessus à l'exception du jeu (les Matches) en lui-même.

Objectifs du projet :

- Mettre au point une base de données cohérente, complète et ouverte à l'extension pour l'ensemble du jeu et de ses composantes (avec anticipation des besoins futurs possibles)
- Mettre au point un système de communication Serveur/Client via requêtes HTTP et fichiers JSON.
- Réaliser une (plusieurs en option) interface (client seulement, la partie administrateur en option) pour les différentes actions (création de compte, connexion, modification d'informations, différentes actions de jeu telles que l'affichage et la manipulation de l'inventaire et du deck).

Etapes du projet

Etude des besoins et conception de la base de données

Trouver et définir toutes les données relatives à la gestion d'un compte, d'un inventaire, de decks, de relations sociales et de matchmaking pour un joueur ; éventuellement certaines données liées à la gestion et à l'administration de la plateforme.

Anticiper au maximum sur la version finale de l'application (contexte bien au-delà des objectifs du projet).

Veiller à la cohérence, à la non-redondance et à l'efficacité des tables et relations.

Création de la base de données

A partir du MCD (Modèle conceptuel de données) réalisé précédemment, écrire les scripts de création des différentes tables du projet.

Choisir les types de chaque attribut et les « range » (valeurs possibles pour chaque attribut) en favorisant l'intégrité de la base et les possibilités d'extension ; une fois la table créée, un nombre minimal de changement doit être effectué sur celles-ci.

Création d'un jeu de données de départ

Ecrire les scripts d'insertion pour certaines tables précises, afin de pouvoir démarrer les tests ainsi que les requêtes sur les tables et relations dépendantes.

Exemple : les tables comme celles des Cartes de Jeu ne dépendent pas (contrairement à la table « compte » par exemple) de l'action d'un client/joueur, elles sont entièrement définies à l'origine par les concepteurs (susceptible d'être modifiées mais le contenu initial est presque final).

Mise en place d'un serveur web avec "container web Tomcat"

Dans l'optique de pouvoir manipuler et traiter les données de la base de données, que ce soit depuis un client Web ou un client Android, nous avons fait le choix de la technologie Tomcat, qui permet d'écrire des « servlets ». Ce sont des fichiers en langage JAVA EE qui sont appelées par le client pour réaliser un traitement précis ; elles sont le parfait intermédiaire entre la requête d'un client, et le résultat basé sur le contenu de la base de données. Voir [Figure 1](#) en Annexe.

Il convient donc, dans un premier temps, d'installer un serveur web (Apache) sur notre machine que l'on appellera alors et pour le reste de ce sujet le « serveur ». On ajoute par-dessus un container web (Tomcat) qui permettra d'appeler des servlets nommées à l'adresse et sur le port configuré.

Analyse des besoins en traitement coté serveur et réalisation (Servlets)

Le serveur Tomcat à présent fonctionnel, il s'agit maintenant de déterminer et concevoir les différents Servlets à implémenter pour répondre à l'ensemble des requêtes du Client.

Des « simples » Servlets de connexion ou de création de compte, aux servlets plus complexes comme celles de l'ouverture d'un « pack de cartes » depuis la boutique ; un ensemble de Servlets est à prévoir en respectant au maximum la sécurité et l'intégrité des données coté serveur ; de plus ces Servlets doivent être ouvert à l'extension autant que possible.

Développement d'une interface web pour client (HTML / CSS / JSP)

La base de données et les servlets étant fonctionnels, le but est alors de pouvoir créer, modifier ou manipuler les données par un client.

Nous réalisons donc une interface web, accessible depuis le navigateur qui permet de réaliser, entre autres, les fonctionnalités suivantes :

- Connexion à un compte
- Création d'un compte
- Modification des attributs d'un compte (email, mot de passe, pseudo, etc..)
- Consultation de l'inventaire (cartes, icones, thèmes, etc..) d'un compte
- Consultation et modification d'un deck (set de cartes pour jouer)
- Consultation et achat d'items dans la boutique du jeu

Développement d'une interface web d'administration (HTML / CSS / JSP)

Le but de cette partie est de concevoir une nouvelle interface, similaire à la précédente, mais cette fois-ci pour répondre aux besoins d'administration. L'accès sera donc réservé aux administrateurs du jeu et leur permettra de, entre autres :

- Modifier, bloquer, bannir, ou supprimer un compte joueur
- Créer et ajouter un item (carte de jeu, icone, thème, etc..)
- Modifier un item existant
- Créer un pack pour la boutique à partir des items existants

Cela permettra de modérer les comptes joueurs, de modifier ou d'ajouter du contenu directement au jeu (à la base de données donc) via une interface claire et simplifiée, bien plus pratique et plus sécurisée que la modification directe de la base de données.