

Octavio : Application Mobile

Le développement de l'application Android a été réalisé à l'aide d'un ensemble d'APIs : Cordova, qui permet le développement d'une application mobile multi-plateforme (Android, iOS, ...) via le développement web.

Pour ce faire, on installera donc Cordova et les différents prérequis comme indiqué sur la page de documentation du site Cordova (rubrique Command Line) : http://cordova.apache.org/docs/en/3.5.0/guide_cli_index.md.html#The%20Command-Line%20Interface

Le développement web se fera dans le dossier "www" au sein du projet, puis celui-ci devra être compilé à l'aide de la commande "cordova build" avant d'être intégré dans Eclipse pour l'exécution Android ou XCode pour iOS.

La suite de ce dossier d'aide présentera notre développement, à savoir le contenu du dossier "www".

1 LES VUES

1.1 LA PAGE PRINCIPALE

Le dossier principal contient les différents fichiers d'affichage.

On y retrouve tout d'abord le fichier index.html, qui contient l'ensemble des informations générales et l'appel à tous les fichiers JavaScript ou CSS. En plus de l'appel à toutes les ressources nécessaires au projet, le fichier index.html contient aussi un élément avec pour id "container", qui contiendra la page.

La page contiendra aussi le chat qui s'affichera ou non selon sa classe CSS (visible / invisible) qui sera modifiée via javascript sur les pages autorisant l'affichage du chat.

La partie avec pour id "container", mentionnée précédemment contiendra un des autres fichiers HTML ajouté via JavaScript en fonction des changements des chargements de page demandés par l'utilisateur.

1.2 LES PAGES DE CONTENU

- home_onlineChoiceView : page initiale, permet de choisir entre le serveur officiel en ligne ou un autre serveur qui peut être sur le même réseau local ou sur internet
- home_serverSelectView : page de configuration du serveur (IP et Port), si on choisit l'option autre serveur dans la page précédente
- home_connectionView : une fois le serveur choisi, cette page permet de se connecter au serveur
- home_subscribeView : page de création de compte sur le serveur, si pas de compte dans la page précédente
- lobbyView : page salon, permet d'accéder au chat, de créer une partie ou de rejoindre une partie existante
- gameFastWaiting : page d'attente, en cas de latence serveur si on lance une partie
- gameConfigView : page de configuration pour une partie personnalisée
- joinGameView : page listant les parties disponibles, pour rejoindre une partie en cours
- gameConfigViewer : page de configuration de partie figée, pour voir les modifications jusqu'au lancement de la partie, toute personne n'étant pas owner s'y rendra jusqu'à ce que l'owner de la partie la démarre
- gameView : page contenant le plateau de jeu

2 LES FICHIERS JAVASCRIPT

Le dossier "script" contient les fichiers JavaScript, ceux-ci sont répartis en 3 parties :

2.1 LA NAVIGATION DANS LES PAGES :

La navigation au sein des pages est assurée par le fichier "switchpage.js", qui met à jour le contenu de la page index via AJAX, afin d'afficher la page demandée par l'utilisateur. Cette page assure aussi le déclenchement du code javascript pour la page de configuration et la page de jeu, pour le code qui doit impérativement être exécuté après le chargement de la page.

La fonction initSwitch() permet le chargement de la première page lors du lancement de l'application.

2.2 LES FICHIERS WEBSOCKET :

Ils possèdent la particule « websocket- » au début de leur nom de fichier et permettent d'identifier les fichiers responsables de l'interaction avec le serveur de jeu.

Ils se regroupent selon la structure suivante :

Un fichier websocket responsable de la connexion avec le serveur.

Un fichier websocket-Index, qui réceptionne les requêtes et les traites.

2.3 LES TRAITEMENTS JS

Les autres fichiers JavaScript vont eux servir aux traitements comme par exemple le calcul du plateau de jeu, le traitement des pages avant l'appel aux web socket comme dans user.js, etc ...

Le fichier « chat » comporte l'ensemble des fonctions utiles au bon fonctionnement du chat (envoi et affichage des messages). On trouve également la fonction toucheEntree() permettant de gérer l'appui sur la touche "Entrée" du clavier (raccourci pour envoyer le message dans le chat, se substituant au bouton "Envoyer"). La gestion de l'affichage des deux chats est également réalisée (on passe du chat général au chat de partie avec l'effet tabs() de jQuery UI).

Le fichier « game » contient l'ensemble des fonctions à réaliser côté client pour la configuration de la partie (création, démarrage, configuration avec construction du JSON, ajout et suppression de lignes dans le formulaire de configuration, ...).

Le fichier "server.js" contient les fonctions liées au choix du serveur et aux modifications nécessaires afin de connecter la websocket au serveur.

Le fichier « user » apporte l'ensemble des fonctions utilisées pour la section membre (connexion, inscription, déconnexion, modification des identifiants, ...).

Le fichier « traitementsboard » comporte un ensemble d'instructions relatives à l'affichage des noms des joueurs au-dessus du plateau de jeu. Il est appelé si le plateau doit s'afficher.

2.4 LE PLATEAU

Le plateau est intégralement codé dans board.js, il utilise la librairie jCanvas qui permet de créer des polygones dans un canvas, de leur assigner des fonctions et de les animer. Il utilise également une image de laurier présente dans le même dossier que le script en question.

Le système pour créer un plateau de jeu à l'écran a été conçu pour être le plus simple possible à utiliser pour une équipe de développement. Ainsi il suffit de sélectionner un canvas avec jQuery, et d'utiliser la fonction d'initialisation du plateau sur ce canvas en renseignant le tour de jeu à afficher en json tel qu'il est envoyé par le serveur, le script s'occupe d'interpréter le json et de créer un plateau interactif et animé dans le canvas.

Cette fonction d'initialisation est une fonction qui implémente les fonctions de jQuery se qui lui permet d'être utiliser comme un plug-in jQuery.

Ainsi on peut également implémenter des fonctions aux éléments entourant le canvas pouvant agir sur le plateau de jeu, on peu par exemple donner des fonctions de zoom et de dézoom à des boutons extérieurs au canvas.

2.5 LES LIBRAIRIES JS

Le fichier « util » permet l'accès à un ensemble de fonctions diverses (traitement des caractères spéciaux, affichage des colonnes, ...)

Le fichier « sha1 » est une librairie de cryptage utilisée pour les mots de passe.

Le dossier « jquery » contient l'ensemble des fichiers nécessaires au bon fonctionnement et affichage des librairies jQuery et jQuery UI.

3 LES FICHIERS DE STYLE CSS & LES RESSOURCES

Les fichiers de style sont stockés dans le dossier "content".

Le dossier ressources comporte les fichiers complémentaires pour le projet, ici le logo.