

# Octavio : Le site

## 1 INDEX

---

Afin de simplifier l'index et d'éviter une répétition, l'index a été divisé en deux parties. La première dans le fichier « index.php » est la page d'entrée au site, elle va définir si la page est rechargée complètement, ou partiellement via un script AJAX.

Dans le cas d'un rechargement complet, les trois parties de la page sont renvoyées, à savoir le menu à gauche, le contenu au milieu et la zone de connexion/chat à droite. Dans le cas d'un rechargement partiel, seul la zone de contenu (ou le cas échéant, la zone de chat) sera renvoyée. Un menu réduit est également chargé. Il se compose de l'ensemble des icônes du menu sur lesquelles on peut cliquer pour accéder aux pages. Ce menu remplace le menu normal si la largeur de la fenêtre est inférieure à 1200 pixels (pour s'adapter aux terminaux mobiles).

Pour permettre de renvoyer le contenu de la page de deux manières simultanément et éviter une redondance du code, le contenu de la page est indexé (on en détermine le contenu) depuis le fichier « indexPages.php ».

On y trouve également deux boutons permettant de masquer manuellement les colonnes de menu et de chat.

Le fichier « .htaccess » contient quant à lui un ensemble d'instructions utilisées pour la compression et la mise en cache des données. Les fichiers compressés sont chargés plus rapidement et la mise en cache permet d'en conserver une partie côté client (et éviter ainsi le rechargement complet de la page à chaque connexion).

Enfin, les requêtes via AJAX comprennent l'ensemble des informations nécessaires à l'index dans l'URL, néanmoins les utilisateurs ne bénéficiant pas de JavaScript peuvent recharger la page et afficher le contenu du site.

Les personnes ne bénéficiant pas de JavaScript ne pourront pas interagir avec le serveur de jeu et ne pourront que consulter le site.

## 2 LES CONTRÔLEURS

---

Le site dispose de deux contrôleurs uniquement, le serveur de jeu se charge de la plus grande partie des traitements.

Le premier est le « pageController », ce fichier est responsable d'effectuer les traitements adaptés à chaque demande, comme récupérer les news, puis afficher la vue correspondant à la requête, ici « newsView ».

Le second contrôleur, « userController », permet en cas de connexion/déconnexion du site, de mettre à jour la zone de chat et de stocker dans les variables de session le nom d'utilisateur ou son absence.

On trouve également un troisième fichier, « pass », qui affecte le mot de passe hashé de l'utilisateur à une variable de session. Cette variable permet à l'utilisateur de se reconnecter automatiquement au rechargement de la page (la fonction de connexion est automatiquement relancée si les variables de session contenant le nom d'utilisateur et le mot de passe existent).

## 3 LES MODÈLES

---

Le dossier Model comprend les managers permettant la lecture des news et des questions/réponses de la FAQ dans la base de données du site, ainsi que les modèles correspondant à la structure de la table, afin de faciliter la manipulation de ces données.

Le dossier comprend aussi un fichier « config.php » qui comprend les informations nécessaires à la connexion à la base de données (localisation, nom, nom d'utilisateur, mot de passe).

## 4 LES VUES

---

### 4.1 LA RÉPARTITION DES VUES

Le dossier « view » contient les différents fichiers d'affichage.

On retrouve ensuite les 3 parties de notre page : le menu, le contenu et le chat.

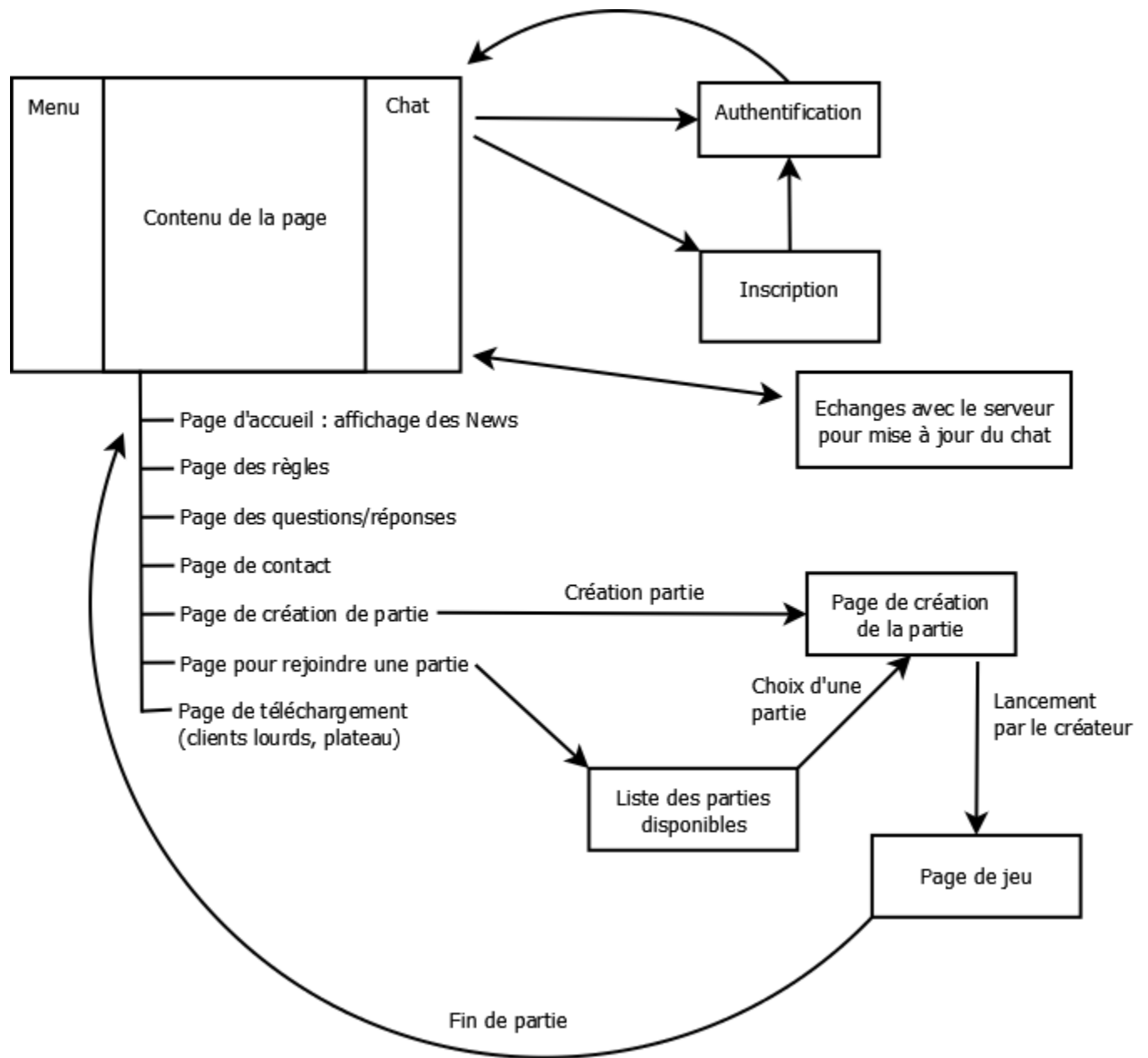
La colonne de menu est affichée par la page « menuView ». On y trouve les différents liens menant vers les pages du site. La page « smallMenuView » comprend les mêmes informations mais l'interface est adaptée aux largeurs de fenêtres/écrans moins grandes.

Le contenu passe par les différentes pages :

Accueil	« newsView »	Créer une partie	« newGameView »
Règles	« rulesView »	Rejoindre une partie	« joinGameView »
FAQ	« faqView »	Inscription	« subscribeView »
Contact	« contactView »		
Téléchargements	« downloadView »		

La page de chat est quant à elle affichée via la page : « chatView ». On y trouve également l'interface de connexion/déconnexion.

## 4.2 DIAGRAMME DES PAGES DU SITE



### 4.3 LES PAGES « SPÉCIALES »

Le fichier « headerView », contient l'ensemble des informations générales (balises meta), l'appel aux feuilles de style, ainsi que l'ouverture du body. « footerView » comprend quant à lui l'appel à l'ensemble des fichiers JavaScript, puis le body se ferme.

Le fichier « siteGlobalVar » comprend deux variables globales utilisées en PHP : le nom du site et le chemin relatif à parcourir pour accéder au site sur le serveur (si le site n'est pas stocké directement à la racine). Le fichier « accountView » contient les interfaces de modification des identifiants (nom et mot de passe). Bien que ces interfaces soient normalement prêtes du côté site, le temps a manqué pour garantir la bonne implémentation des traitements côté serveur. L'affichage de ces formulaires est donc désactivé.

Le fichier « gameConfigView » contient le formulaire de configuration de la partie de base. Pour une partie normale, le minimum jouable est déjà affiché (deux teams, un joueur par team, une légion par joueur). On peut ajouter teams, joueurs et légions à loisir, modifier le temps d'un tour et la taille du plateau. Pour une partie rapide, seule la mention "En attente d'un joueur" est affichée. Le fichier « gameConfigViewer » contient le formulaire de configuration de partie pour les invités d'une salle, mis à jour mais non modifiable.

Enfin, le fichier « gameView » comprend la structure d'affichage du plateau, avec un div contenant les noms des joueurs et leurs pions associés, les boutons de zoom du plateau, la barre indiquant le temps restant avant la fin du tour, les boutons permettant de finir le tour et de quitter la partie, et le canvas contenant le jeu.

## 5 LES FICHIERS JAVASCRIPT

---

Le dossier « js » contient les fichiers JavaScript, ceux-ci sont répartis en 3 parties :

### 5.1 LES FICHIERS AJAX :

Ils sont repérables par le préfixe « AJAX- ». Ces fichiers déclenchent des requêtes asynchrones afin de mettre à jour le contenu de la page principale ou du chat.

### 5.2 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 websocketIndex, qui réceptionne les requêtes et les traite (gestion des messages d'erreur, affichage de la configuration de partie, gestion de fin de partie, affichage des utilisateurs, ...)

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

Le fichier « load » recharge le contenu des liens afin d'adapter les redirections si le visiteur n'a pas désactivé JavaScript.

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.

Le fichier « global » contient un ensemble de variables globales JavaScript utilisées par les fonctions du site.

### 5.4 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.

### 5.5 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.

## 6 LES FICHIERS DE STYLE CSS

---

L'intégralité des feuilles de style du site est placée dans le dossier « css » du site. « chat » contient le style de la colonne de chat, « menu » apporte l'affichage de la colonne de gauche, « main » donne le reste du style appliqué au site.

La police d'écriture Ubuntu-R est utilisée sur le site, son fichier .ttf est accessible [ici](#).

Enfin, le dossier « images » comporte l'ensemble des images utilisées sur le site.