

# Mini-projet 2017 – 2018 DUT 2

Par Victor BOIX et Evann BACALA

Travail réalisé dans le cadre du module M3104



UNIVERSITÉ DE NANTES

## Table des matières

Introduction.....	2
Architecture des scripts.....	3
Index.....	3
Contrôleur .....	4
Modèle .....	4
Vue .....	4
Techniques utilisées .....	5
Champs cachés .....	5
Variables de session .....	5
Conclusion .....	6

## Introduction

Dans le cadre du module de Programmation Web Serveur nous devons réaliser le jeu du solitaire en langage PHP. L'objectif du jeu est de supprimer des billes du plateau afin qu'à la fin il n'en reste plus qu'une. Pour qu'une bille soit ôtée du plateau, on doit sauter par-dessus avec une autre bille en direction d'une case vide. La prise en diagonale est interdite.

Nous devons le concevoir en suivant l'architecture MVC (Modèle Vue Contrôleur). Ce modèle consiste à ce que le code soit construit de la façon suivante :

- Un routeur qui sert à rediriger les requêtes de l'utilisateur vers le bon contrôleur
- Le Modèle qui contient le code du programme
- Des Vues avec lesquelles l'utilisateur va pouvoir interagir
- Les Contrôleurs qui servent d'interfaces entre le Modèle et les Vues

## Architecture des scripts

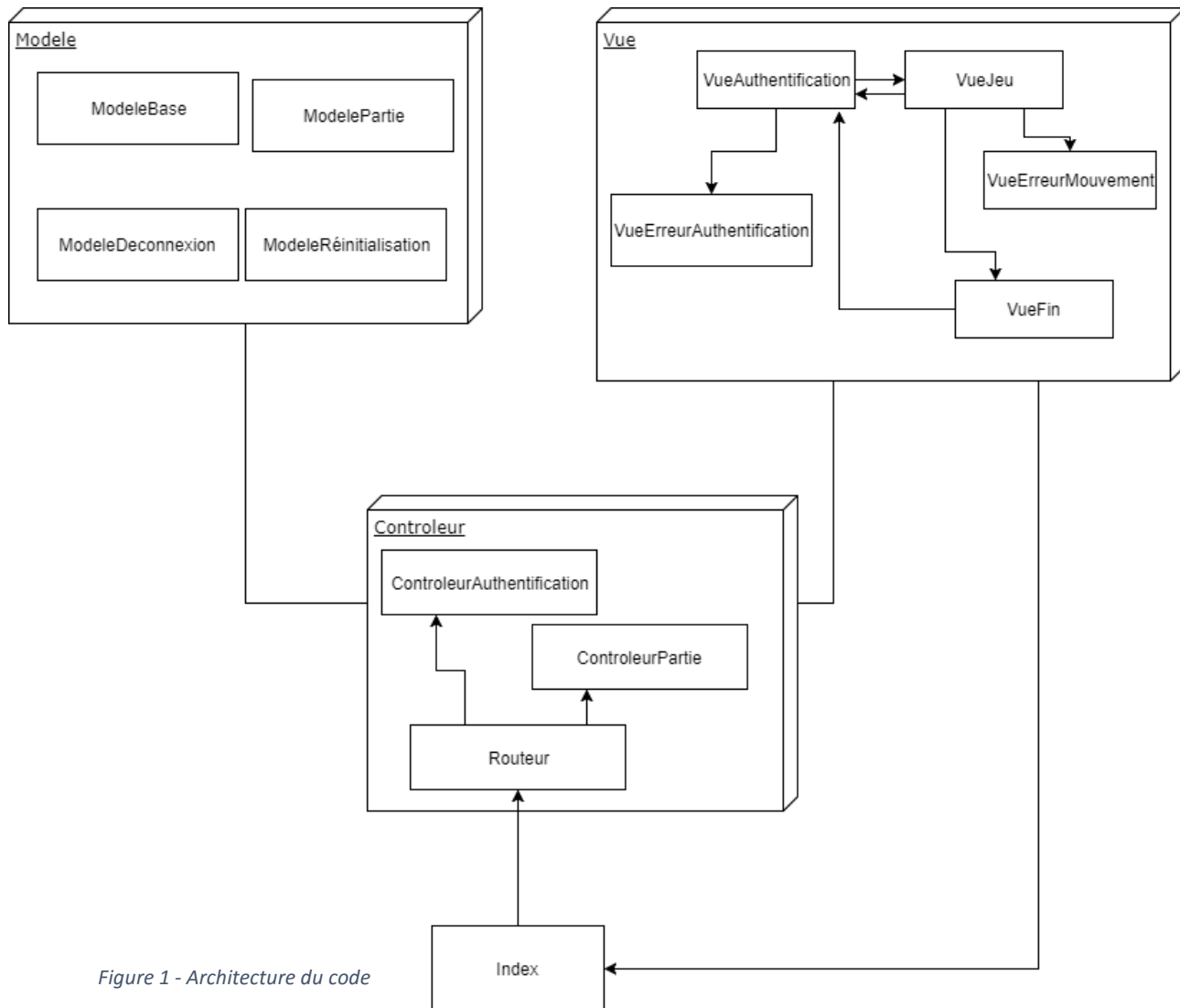


Figure 1 - Architecture du code

## Index

Le script `index.php` est le fichier à la base du code, c'est lui qui va ouvrir la session, il va créer un routeur puis va appeler sa méthode `routerRequête`.

## Contrôleur

Le routeur est appelé par l'index et va servir à router les requêtes de l'utilisateur (pour plus d'informations sur son fonctionnement voir les commentaires du code du routeur).

Le contrôleur d'authentification (ControleurAuthentification) va vérifier si les identifiants et mots de passe sont corrects, si c'est le cas alors il renvoie true sinon il renvoie false.

Le contrôleur de partie (ControleurPartie) va lui s'occuper de gérer le jeu et contient donc les méthodes qui permettent de déplacer un pion, vérifier qu'un mouvement est valide, supprimer un pion, vérifier la victoire ou la défaite, créer une nouvelle partie et enfin afficher la vue de jeu (VueJeu).

## Modèle

Le script ModeleBase va comporter toutes les méthodes qui permettent d'interagir avec la base de données tel que : ouvrir une connexion, fermer la connexion, récupérer les pseudos, récupérer un mot de passe, vérifier qu'un pseudo existe ainsi que récupérer les 3 meilleurs joueurs avec leurs statistiques.

Le script ModelePartie va quant à lui contenir tout le code qui permet d'instancier et récupérer et modifier une partie.

Le script ModeleDeconnexion lui va contenir le code qui permet de se déconnecter de la session courante.

Le script ModeleReinitialisation va contenir le code qui permet de réinitialiser une partie.

## Vue

La première vue que l'utilisateur va rencontrer est la vue d'authentification (VueAuthentification) qui va demander un login et un mot de passe (il n'existe dans la base que 2 joueurs « titi » et « toto » avec comme mot de passe respectifs « titi » et « toto »).

Si l'identification est invalide (mauvais login ou mot de passe) alors une vue va s'afficher (VueErreurAuthentification) en indiquant que l'identification est invalide avec un lien permettant de revenir à la vue d'authentification.

Si l'authentification est correcte alors la vue du jeu (VueJeu) s'affiche et va contenir un plateau avec des pions (représentés par des cookies), une case blanche et des cases noires, qui sont des cases injouables. La vue va aussi contenir un lien de déconnexion qui va renvoyer à la vue d'authentification ainsi qu'un lien qui permet de réinitialiser la partie. Pour jouer l'utilisateur sélectionne un cookie puis il doit sélectionner une case blanche à une case de distance et pas en diagonal. Le cookie alors sauté est supprimé et le cookie qui l'a sauté a changé de position.

Si le mouvement effectué par le joueur est invalide alors une vue (VueErreurMouvement) indiquant que le mouvement est invalide est affiché avec un lien permettant de revenir à la vue du jeu. Idem si le joueur sélectionne une case blanche en premier.

Si le joueur gagne la partie (plus qu'un cookie sur le plateau) alors une vue s'affiche (VueFin). Il est affiché que le joueur a gagné la partie ainsi que la liste des 3 meilleurs joueurs. Il y a aussi un lien pour recommencer une nouvelle partie ainsi qu'un lien pour se déconnecter.

Si le joueur perd la partie (si tous les pions restant se retrouvent avec que des cases blanches autour d'eux (est, ouest, nord et sud) alors la même vue de fin s'affiche à la différence qu'il est écrit que le joueur a perdu.

## Techniques utilisées

En ce qui concerne les techniques que nous avons utilisés. Nous avons tout d'abord essayé de programmer en objet. Cela nous est particulièrement profitable pour créer des instances de Modèles, de Contrôleurs et de Vues. De manière générale ce mode de programmation est très adapté à la création de jeux.

### Champs cachés

Pour simuler le clic sur les cases du plateau nous avons opté pour l'utilisation de formulaires avec champs cachés qui renvoient en méthode post au routeur les coordonnées de la case cliquée.

### Variables de session

La méthode que nous avons le plus utilisé afin de sauvegarder des données sont les variables de session. En effet nous avons stocké en variable de session les coordonnées de la case de départ d'un pion ainsi que les coordonnées de la case d'arrivée afin d'être réutilisée dans le contrôleur de la partie.

L'instance courante de la partie a elle aussi été passé en variable de session. Car à chaque appel du routeur, celui-ci récrée une nouvelle instance du contrôleur de partie qui lui crée une nouvelle instance de partie. On a donc décidé de passer la partie en variable de session pour qu'elle ne soit pas réinitialisée à chaque rechargement de la page.

## Conclusion

Bien que l'on ait rencontré des difficultés. Notre avis général sur ce projet est très positif car il nous a permis de réaliser de gros progrès en développement Web. En effet bien que l'on n'ait pas fait de css pour rendre notre site un peu plus agréable à l'œil, il nous a permis d'exploiter pleinement les capacités PHP mais aussi d'expérimenter ses limites.

En ce qui concerne nos perspectives d'évolution pour ce programme nous voudrions tous d'abord le rendre plus esthétique en ajoutant du css, rajouter une option pour permettre de créer un joueur et enfin optimiser l'exécution du code.

Nous serions très intéressées par un autre projet de ce type mais cette fois ci en javascript pour explorer le domaine de la programmation Web côté client.