

Atelier 1 : évaluation Prog. Web / Serveur

1. Dans le cadre du projet, nous avons principalement utilisé le micro-framework Slim. Celui-ci a l'avantage de générer à lui seul l'ensemble du système MVC. Néanmoins nous lui avons adjoint le moteur de templates Twig pour combler certaines lacunes de Slim.

2. Slim présente la particularité de ne pas présenter de mécanisme de layout à proprement parlé, puisqu'il effectue ses réponses sous forme de réponses HTTP. C'est pour cela que nous avons utilisé Twig en parallèle, comme cela est conseillé par la documentation Slim.

Twig propose une syntaxe assez simple pour créer du code dynamique, plus simple que du PHP. De plus il permet d'utiliser l'héritage de templates, ce qui réduit considérablement le temps de code.

Dans le cas d'un jeu de mastermind, j'utiliserai des templates partielles. Ils seraient appelé par la fonction correspondant au jeu avec comme paramètre le coup joué :

<http://monmastermind.com/game?1=red&2=yellow&3=green&4=yellow>

3. Revue de code : morpion (PHP)

Dans un premier temps, la fonction définit ce que sont les deux diagonales :

```
$diagonals = [  
    [$this->board[0][0], $this->board[1][1], $this->board[2][2]]  
    [$this->board[0][2], $this->board[1][1], $this->board[2][0]]  
];
```

[0][0]		[2][0]
	[1][1]	
[0][2]		[2][2]

Ce sont deux tableaux qui correspondent aux positions des diagonales.

Ensuite pour chacune des deux diagonales, on va compter le nombre de valeurs contenues dans les cases du tableaux correspondant à la diagonale. Cette somme sera contenue sous forme de dans la variable \$scores :

```
foreach($diagonals as $diagonal){  
    $scores = array_count_values(array_filter($diagonal));
```

Puis s'il existe une clef du nom correspondant au \$playerId dans le tableau \$scores et si la valeur correspondante à cette clef est égale à 3, auquel cas la fonction retournera True car le joueur aura bien réussi à aligner une colonne :

```
if(
    array_key_exists($playerId, $scores) && $scores[$playerId] == 3
){
    return true;
}
```

Si on sort de la boucle du foreach sans avoir trouvé de diagonale alignée, la fonction retourne False.

La condition du if me paraît assez discutable. En effet on vérifie si la clef existe ET si elle correspond à la valeur 3. On pourrait simplement utiliser `$scores[$playerId] == 3`. Si la clé `$playerId` n'est pas présente dans le tableau, on sortira immédiatement du if.

4. Un routeur est la partie d'une application web qui va faire correspondre une requête HTTP avec un handler, typiquement une fonction ou une méthode, en fonction de la méthode utilisé et du chemin spécifié dans l'URL.

Pour le jeu de bataille, les routes possibles sont :

Route : /

Description : renvoie à la page d'accueil du site

Code de déclaration de la route : `get('/', ['as' => 'home', function() { ...}]);`

Exemple d'appel : <http://mabataille.com/>

Route : /waiting

Description : correspond à la page d'attente lors d'une recherche de partie

Code de déclaration : `post('/waiting', ['as' => 'waiting', function() { ...}]);`

Exemple d'appel : <http://mabataille.com/waiting>

Relation : lorsqu'une partie est trouvée, elle va automatiquement rediriger vers /game

Route : /game

Description : correspond à la page de la partie jouée, avec son id

Code de déclaration : `get('/game', ['as' => 'game', function() { ...}]);`

Exemple d'appel : <http://mabataille.com/game?id=45456>

Relation : lorsque la partie est terminée, elle va rediriger vers /

5. Une faille XSS est une faille de sécurité qui permet à un utilisateur malveillant d'injecter du code dans une page web.

La façon la plus simple de s'en protéger est d'utiliser la fonction `htmlspecialchars()`. Celle-ci va filtrer et remplacer les symboles potentiellement problématiques, typiquement des symboles comme `&`, `>` ou `'`, en équivalant HTML.