

# Mission 3 : Agora Web (Utilisation de Twig)

**Valentin, Yenis, Marco, Marvin**

Projet réalisé : Transformation du système de vue pour le site d'administration de la MJC AGORA de Libreville (site créé durant la Mission 1)

## Contexte mission

Après la première mission où nous avons développé le site d'administration de la **MJC AGORA de Libreville** en PHP avec une architecture **MVC classique**, nous avons poursuivi avec une seconde étape du projet : **l'intégration de Twig comme moteur de templates**.

L'objectif de cette évolution était d'améliorer la séparation entre la logique et la présentation, d'avoir un code plus lisible et maintenable, et de se rapprocher des standards actuels du développement web.

## Organisation

Trello, Github, GDoc pour les CR : chaque étape est définie sur Trello pour avoir une vision globale de l'avancée de la mission afin de s'organiser le mieux possible.. Pour Github nous utilisons la méthode du "pusher" pour éviter les conflits lors des push, Marvin à ce rôle dans le groupe.

### **Nous nous sommes partagés ainsi les tâches :**

Implantation de Twig (dossiers et index.php): **Tout le groupe**

Implantation de Twig dans Jeux : **Valentin**

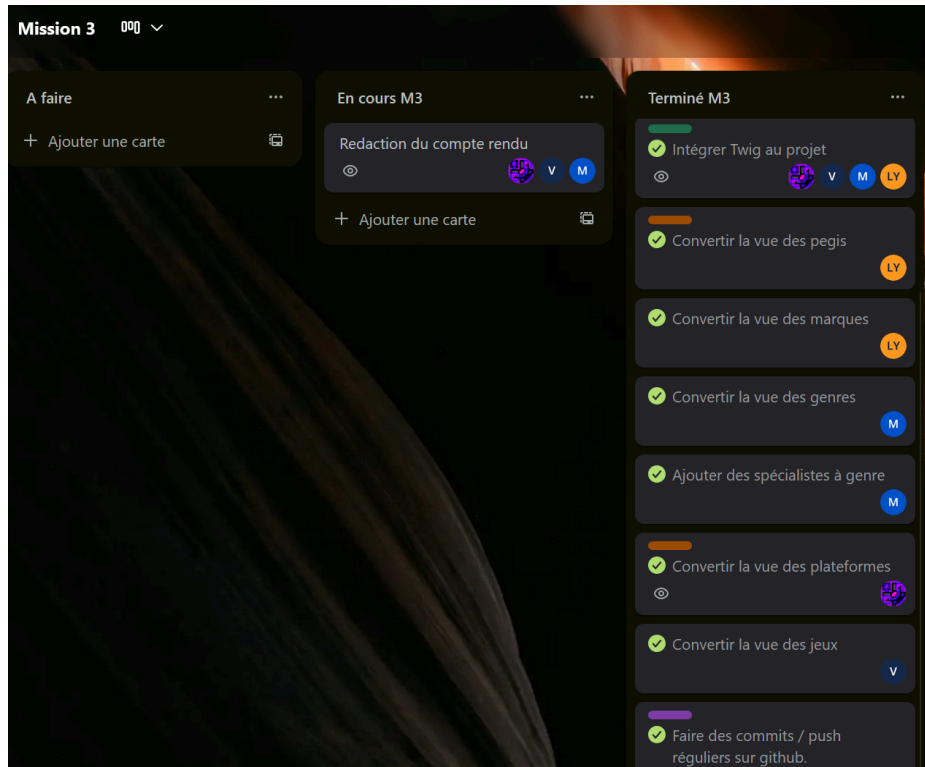
Implantation de Twig dans Marques : **Yenis**

Implantation de Twig dans Pegis : **Yenis**

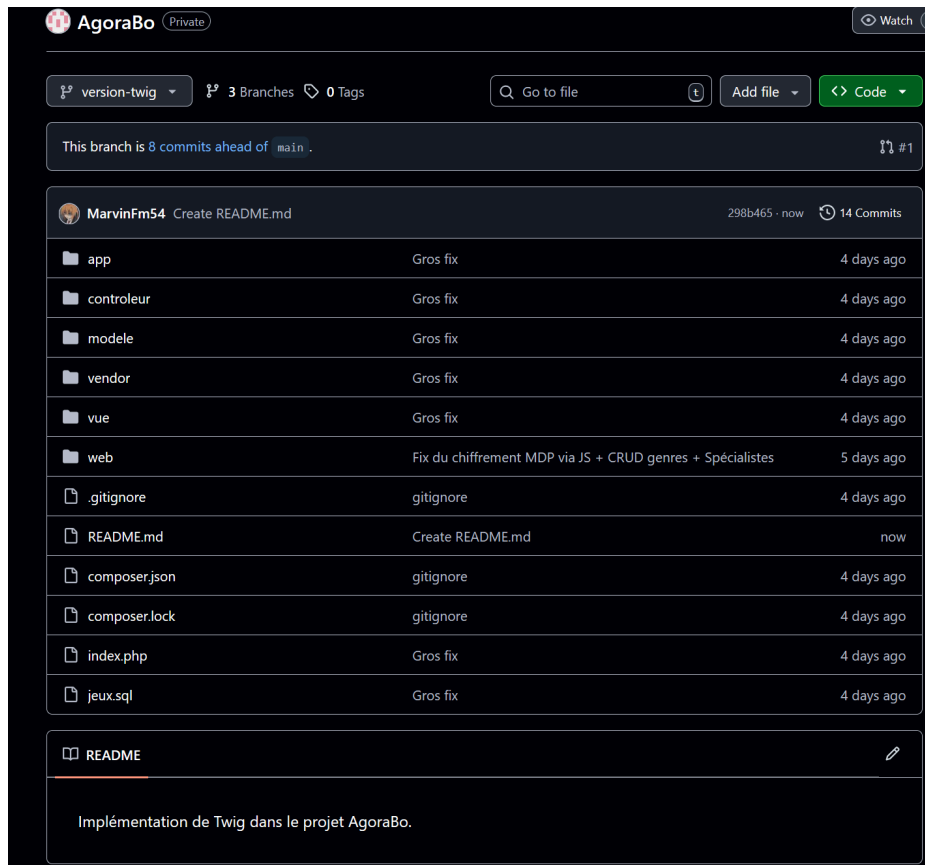
Implantation de Twig dans Genres : **Marco**

Implantation de Twig dans Plateformes : **Marvin**

## Capture d'écran du Trello



## Capture d'écran de la branche GitHub



## Technologies utilisées

- **Back-end** : PHP
- **Front-end** : Twig, HTML, Bootstrap, JS
- **Base de données** : MySQL
- **Architecture** : MVC
- **Outils** : Visual Studio Code, GitHub, Apache, Xampp/Wamp

## Pourquoi Twig ?

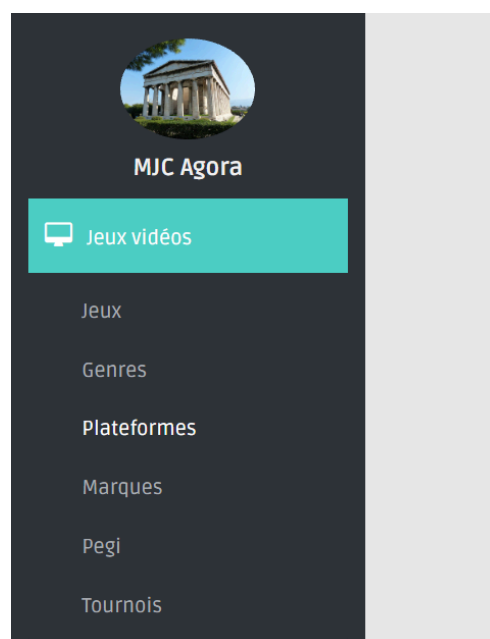
Twig permet de séparer clairement la logique et l'affichage, rendant le code plus lisible et maintenable. Sa syntaxe simple facilite l'affichage des données, l'héritage de templates favorise la réutilisation, et l'échappement automatique renforce la sécurité.

## Cas d'utilisation

### Affichage des plateformes avec Twig :

- L'utilisateur (admin) choisit « Plateformes » dans le menu. index.php appelle le contrôleur c\_gererPlateformes.php.

```
case 'gererPlateformes': {  
    require 'contrôleur/c_gererPlateformes.php';  
    break;  
}
```



- Le contrôleur récupère les données via le modèle (getLesPlateformes() dans class.PdoJeu.inc.php).

```
public function getLesPlateformes(): array
{
    $requete = 'SELECT idPlateforme as identifiant, libPlateforme as libelle
              FROM plateforme
              ORDER BY libPlateforme';

    try {
        $resultat = PdoJeu::$monPdo->query(query: $requete);
        $tbPlateformes = $resultat->fetchAll();
        return $tbPlateformes;
    } catch (PDOException $e) {
        die('<div class = "erreur">Erreur dans la requête !<p>'
            . $e->getMessage() . '</p></div>');
    }
}
```

- Les données sont transmises à un template Twig (lesPlateformes.html.twig).

```
lesPlateformes.html.twig
{% extends "layout.html.twig" %}
{% block central %}
<div class="col-sm-12">
<section class="panel">
<div class="chat-room-head">
<h3>
<i class="fa fa-angle-right"></i>
Gérer les plateformes
</h3>
</div>
<div class="panel-body">
<table class="table table-striped table-advance table-hover">
<thead>
<tr class="tableau-entete">
<th>
<i class="fa fa-bullhorn"></i>
Identifiant</th>
<th>
<i class="fa fa-bookmark"></i>
Libellé</th>
</tr>
</thead>
<tbody>
<!-- Nouveau -->
<tr>
<td>
<form action="index.php?uc=gererPlateformes" method="post">
<td>Nouveau</td>
<td>
<input type="text" id="txtLibPlateforme" name="txtLibPlateforme">
</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
```

- Twig affiche proprement la liste des plateformes avec une boucle {% for plateforme in tbPlateformes %}.

```
{% for plateforme in tbPlateformes %}
<tr>
<td>
<form action="index.php?uc=gererPlateformes" method="post">
<td>
{{ plateforme.identifiant }}
<input type="hidden" name="txtIdPlateforme" value="{{
plateforme.identifiant }}" />
</td>
<td>
{{ plateforme.libelle }}
<input type="text" id="txtLibPlateforme" name="txtLibPlateforme" value="{{
plateforme.libelle }}" />
</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
```

