

# ShareGames

---

Documentation technique

AVRIL-MAI 2023

Billegas Lucas

I.DA-P4B

Maître d'apprentissage : M. Garchery



## Table des matières

<b>TABLE DES MATIÈRES.....</b>	<b>3</b>
<b>VERSIONS .....</b>	<b>5</b>
<b>INTRODUCTION .....</b>	<b>5</b>
<b>RÉSUMÉ DU CAHIER DES CHARGES.....</b>	<b>5</b>
<i>But de l'application .....</i>	<i>6</i>
<i>Le programme consiste à mettre en relation des utilisateurs par rapport à leurs intérêts pour des jeux vidéo.....</i>	<i>6</i>
<i>Les administrateurs ont la charge de saisir et gérer les fiches des jeux vidéo et les associés à des mots clefs (type de jeux) et un visuel.....</i>	<i>6</i>
<i>Les utilisateurs peuvent noter et commenter ces jeux sous forme de billets. ....</i>	<i>6</i>
<i>Le programme affiche ensuite le classement des jeux suivant leur note et leur intérêt qu'il suscite auprès de la communauté. ....</i>	<i>6</i>
<i>Fonctionnalités à réaliser .....</i>	<i>6</i>
<i>Matériel et logiciels Nécessaires .....</i>	<i>7</i>
<i>Livrables .....</i>	<i>7</i>
<b>MÉTHODOLOGIE.....</b>	<b>8</b>
<i>S'informer.....</i>	<i>8</i>
<i>Planifier .....</i>	<i>8</i>
<i>Décider .....</i>	<i>8</i>
<i>Réaliser.....</i>	<i>8</i>
<i>Contrôler .....</i>	<i>8</i>
<i>Evaluer .....</i>	<i>8</i>
<b>PLANIFICATION.....</b>	<b>9</b>
<i>Product backlog .....</i>	<i>9</i>
<i>Planning prévisionnel.....</i>	<i>14</i>
<i>Planning réel .....</i>	<i>15</i>
<b>ANALYSE FONCTIONNELLE .....</b>	<b>16</b>
<i>Fonctionnalités.....</i>	<i>16</i>
<i>Structure du projet .....</i>	<i>24</i>
<i>Base de données.....</i>	<i>25</i>
<b>LIBRAIRIES ET OUTILS EXTERNES .....</b>	<b>26</b>
<i>Bootstrap.....</i>	<i>26</i>
<i>Visual Studio Code.....</i>	<i>26</i>
<i>WSL 2 (Ubuntu) .....</i>	<i>26</i>
<i>Pecce/simple-router.....</i>	<i>26</i>
<i>Github.....</i>	<i>27</i>
<i>Composer .....</i>	<i>27</i>
<i>Autoload.....</i>	<i>27</i>
<b>PÉRIMÈTRE DE TEST.....</b>	<b>28</b>
<b>ÉQUIPEMENT DE TEST .....</b>	<b>28</b>
<b>PLAN DE TEST .....</b>	<b>29</b>

<b>ÉVOLUTION DES TESTS.....</b>	<b>35</b>
<b>RAPPORT DE TEST .....</b>	<b>36</b>
<b>CONCLUSION .....</b>	<b>43</b>
<i>Planification .....</i>	<i>43</i>
<i>Améliorations possibles.....</i>	<i>43</i>
<i>Bilan personnel.....</i>	<i>43</i>
<i>Remerciements.....</i>	<i>43</i>
<b>ANNEXES.....</b>	<b>44</b>
BIBLIOGRAPHIE.....	44
<i>Template .....</i>	<i>44</i>
<i>Sites web utilisés .....</i>	<i>44</i>
INSTALLATION .....	50
<i>Composer .....</i>	<i>50</i>
<i>Simple-router.....</i>	<i>50</i>
<i>Autoload.....</i>	<i>50</i>
MISE EN PLACE D'UN VIRTUAL HOST.....	51
<i>Création.....</i>	<i>51</i>
<i>apache2.conf.....</i>	<i>51</i>
<i>Hosts.....</i>	<i>51</i>
CONFIGURATION FICHIERS AUTOLOAD ET PECEE/SIMPLE-ROUTER.....	53
<i>autoload-psr4.php.....</i>	<i>53</i>
<i>autoload_static.php.....</i>	<i>53</i>
<i>.htaccess.....</i>	<i>54</i>
<i>composer.json .....</i>	<i>54</i>
<i>index.php.....</i>	<i>55</i>
<i>routes.php .....</i>	<i>55</i>
CODE SOURCE .....	57
<i>index.php.....</i>	<i>57</i>
<i>routes.php .....</i>	<i>57</i>
<i>composer.json .....</i>	<i>58</i>
<i>config-exemple.php.....</i>	<i>58</i>
<i>.htaccess.....</i>	<i>59</i>
<i>Models.....</i>	<i>60</i>
<i>Controllors .....</i>	<i>75</i>
<i>Views .....</i>	<i>100</i>

## Versions

N. de version	Auteur	Date	Changements
1.0	Billegas Lucas	27.04.2023	Maquette
2.0		11.05.2023	Intermédiaire
3.0		17.05.2023	Finale

## Introduction

Ce document est un rapport décrivant toute la conception du projet réalisé dans le cadre du Travail pratique individuel (TPI) se déroulant du 27 avril au 17 Mai 2023. Ce travail a pour but de valider mes compétences techniques acquises durant toute ma formation d'informaticien en développement d'applications.

## Résumé du cahier des charges

Les informations ci-dessous sont extraites du cahier des charges du TPI :

Élève	Maître d'apprentissage
Billegas Lucas <a href="mailto:lucas.blgs@eduge.ch">lucas.blgs@eduge.ch</a>	Garchery Stéphane <a href="mailto:stephane.garchery@edu.ge.ch">stephane.garchery@edu.ge.ch</a>

Experts	
Bouille Robin <a href="mailto:robin.bouille@gmail.com">robin.bouille@gmail.com</a>	Bizard Yohan <a href="mailto:yohan.bizard@gmail.com">yohan.bizard@gmail.com</a>

## But de l'application

Le programme consiste à mettre en relation des utilisateurs par rapport à leurs intérêts pour des jeux vidéo.

Les administrateurs ont la charge de saisir et gérer les fiches des jeux vidéo et les associés à des mots clefs (type de jeux) et un visuel.

Les utilisateurs peuvent noter et commenter ces jeux sous forme de billets.

Le programme affiche ensuite le classement des jeux suivant leur note et leur intérêt qu'il suscite auprès de la communauté.

## Fonctionnalités à réaliser

### **Administrateur :**

Un administrateur peut créer, modifier, supprimer et voir une fiche de jeu.

Un administrateur peut supprimer un commentaire d'un utilisateur.

### **Utilisateurs enregistrés :**

Un utilisateur enregistré peut créer, modifier, supprimer et voir son commentaire sur une fiche de jeu.

Un utilisateur enregistré peut voir toutes les fiches de jeux ou il a posté un ou plusieurs commentaires.

Un utilisateur enregistré peut modifier son mot de passe et son avatar.

Un utilisateur enregistré peut voir son profil.

### **Utilisateurs non-enregistrés :**

Un utilisateur non-enregistré peut voir la liste des 10 meilleurs jeux en fonction de leur moyenne sur les notes utilisateurs.

Un utilisateur non-enregistré peut voir les 5 dernières fiches de jeux publiées ou/et mise à jour.

Un utilisateur non-enregistré peut rechercher une fiche par la description ou le titre.

Un utilisateur non-enregistré peut voir toutes les fiches de jeux et les filtrer par date de création / modification ou par la moyenne des notes.

## Matériel et logiciels Nécessaires

- 1 PC standard école
- Windows 10
- WSL
- Suite MSOffice

## Livrables

- Rapport de projet
- Manuel utilisateur
- Résumé du rapport du TPI
- Journal de travail

# Méthodologie

Pour la gestion de mon projet j'ai décidé d'utiliser la méthode de planification en 6 étapes. Voici une petite description de ces étapes.

## S'informer

Se renseigner est la première des 6 étapes de la méthodologie. Elle consiste à comprendre les différentes attentes et à bien cerner le sujet afin de se faire une idée de l'objectif et se procurer systématiquement les bonnes informations.

## Planifier

Planifier est le meilleur moyen d'accomplir son travail dans un temps défini. Pour ce faire, il faut transformer les fonctionnalités demandées en tâches et estimer le temps nécessaire pour effectuer chacune d'entre elles.

## Décider

Après avoir récupéré les informations et planifié les tâches, il faut décider de l'importance de chaque tâche, ce qui permet d'avoir un ordre de réalisation.

## Réaliser

Une fois les tâches triées par importance, il faut réaliser ces dernières. Pour cela, il faut suivre la planification et les décisions qui ont été faites précédemment.

## Contrôler

Une fois la tâche terminée, il faut vérifier qu'elle fonctionne correctement. Il faut donc mettre en place un scénario de test afin de tester efficacement la fonctionnalité qui a été implémentée.

## Evaluer

La tâche étant terminée, il faut passer en revue cette dernière afin de voir comment elle a été faite, s'il y a des choses qui laissent à désirer ainsi que ce qui pourrait être amélioré pour les prochaines tâches.

# Planification

## Product backlog

Nom	<b>S1 : Utilisation d'un dépôt Git</b>
Description (user story)	En tant que développeur, je dois utiliser un dépôt Git afin d'enregistrer et versionner mon projet.
Critère d'acceptation	Un dépôt Git est créé et est en lien avec GitHub Desktop.
Priorité	P1 : Bloquant

Nom	<b>S2 : Installation de la base de données</b>
Description (user story)	En tant que développeur, je dois installer une base de données qui respecte le MCD qui m'a été donné dans l'énoncé.
Critère d'acceptation	La base de données est créée ainsi que les tables et un user spécifique à cette base de données.
Priorité	P1 : Bloquant

Nom	<b>S3 : Création du style du site</b>
Description (user story)	En tant que développeur, je dois créer le visuel de chaque page.
Critère d'acceptation	Le visuel de chaque page correspond à la description de l'énoncé.
Priorité	P1 : Bloquant

Nom	<b>S4 : Création d'une fonction PHP pour la connexion à la base de données</b>
Description (user story)	En tant que développeur, je dois créer une connexion à la base de données pour pouvoir récupérer les données.
Critère d'acceptation	Les données peuvent être récupérées, ajoutées, supprimées et éditées depuis des fonctions PHP.
Priorité	P1 : Bloquant



Nom	<b>S5 : Installation et configuration de composer et des librairies externes</b>
Description (user story)	En tant que développeur, je dois mettre en place un routeur pour gérer les url et mieux gérer le code.
Critère d'acceptation	Le test 1.1 passe.
Priorité	P1 : Bloquant

Nom	<b>S6 : Mise en place d'un Virtual Host</b>
Description (user story)	En tant que développeur, je dois créer un Virtual host qui me permettra de mettre le routeur dessus.
Critère d'acceptation	Le site est fonctionnel depuis un navigateur.
Priorité	P1 : Bloquant

Nom	<b>S7 : Affichage des jeux sur la page d'accueil</b>
Description (user story)	En tant qu'utilisateur, je dois voir la liste des 10 meilleurs jeux selon la moyenne de leurs notes. Je dois aussi pouvoir voir les 5 jeux les plus récents selon leur date de saisie / modification.
Critère d'acceptation	Le test 2.1 passe.
Priorité	P2 : Critique

Nom	<b>S8 : Recherche de jeux</b>
Description (user story)	En tant qu'utilisateur, je dois pouvoir rechercher des jeux par le titre ou la description.
Critère d'acceptation	Le test 3.1 passe.
Priorité	P2 : Critique

Nom	<b>S9 : Affichage et filtrage des jeux sur une page</b>
Description (user story)	En tant qu'utilisateur, je dois pouvoir filtrer les jeux par date de saisie / modification ou par note.
Critère d'acceptation	Les tests 4.1 et 4.2 passent.
Priorité	P2 : Critique

Nom	<b>S10 : Inscription</b>
Description (user story)	En tant qu'utilisateur, je dois pouvoir m'inscrire en saisissant un pseudo, un mot de passe et un avatar en option.
Critère d'acceptation	Le test 5.1 passe.
Priorité	P1 : Bloquant

Nom	<b>S11 : Connexion</b>
Description (user story)	En tant qu'utilisateur, je dois pouvoir me connecter à mon compte avec mes informations.
Critère d'acceptation	Le test 6.1 passe.
Priorité	P1 : Bloquant

Nom	<b>S12 : Gestion des fiches de jeux</b>
Description (user story)	En tant qu'administrateur, je peux ajouter, supprimer, voir et modifier une fiche de jeu.
Critère d'acceptation	Les tests 7.1 à 7.3 passent.
Priorité	P2 : Critique

Nom	<b>S13 : Déconnexion</b>
Description (user story)	En tant qu'utilisateur identifié, je peux me déconnecter de mon compte.
Critère d'acceptation	Le test 8.1 passe.
Priorité	P4 : Secondaire

Nom	<b>S14 : Gestion des avis</b>
Description (user story)	En tant qu'utilisateur identifié, je peux modifier, supprimer, voir et créer un avis.
Critère d'acceptation	Les tests 9.1 à 9.3 passent.
Priorité	P2 : Critique

Nom	<b>S15 : Suppression des avis</b>
Description (user story)	En tant qu'administrateur, je peux supprimer un avis.
Critère d'acceptation	Le test 10.1 passe.
Priorité	P3 : Important

Nom	<b>S16 : Page profile</b>
Description (user story)	En tant qu'utilisateur identifié, je peux modifier mon avatar et mon mot de passe. J'ai aussi accès à toutes les fiches de jeux dans lesquelles j'ai posté un ou plusieurs avis.
Critère d'acceptation	Les tests 11.1 à 11.3 passent.
Priorité	P3 : Important

Nom	<b>S17 : Détails des fiches de jeux</b>
Description (user story)	En tant qu'utilisateur, je peux voir une fiche de jeux avec ses commentaires trier par ordre chronologique sous la fiche.
Critère d'acceptation	Le test 12.1 passe.
Priorité	P3 : Important

Nom	<b>S18 : Mot de passe oublié</b>
Description (user story)	En tant qu'utilisateur, si j'oublie mon mot de passe, je peux cliquer sur un lien qui me permet de générer un nouveau mot de passe.
Critère d'acceptation	Le test 13.1 passe.
Priorité	P3 : Important

Nom	<b>S19 : Pagination page jeux</b>
Description (user story)	En tant qu'utilisateur, j'ai accès à une pagination qui me permet de naviguer entre les différentes pages qui contiennent des jeux. Si j'ai une recherche ou un filtre il doit être sauvegarder entre les différentes pages.
Critère d'acceptation	Le test 14.1 passe.
Priorité	P4 : Secondaire

Nom	<b>S20 : Pagination page gestion</b>
Description (user story)	En tant qu'administrateur, j'ai accès à une pagination qui me permet de naviguer entre les différentes pages qui contiennent des jeux.
Critère d'acceptation	Le test 14.2 passe.
Priorité	P4 : Secondaire

# Planning prévisionnel

Le planning prévisionnel permet d'avoir des objectifs. C'est des prévisions, voilà pourquoi il y a aussi un planning effectif

		27. avr 2023	2. mai 2023	3. mai 2023	4. mai 2023	8. mai 2023	9. mai 2023	10. mai 2023	11. mai 2023	15. mai 2023	16. mai 2023	17. mai 2023
Tâches à réaliser	Temps nécessaire											
<b>S'informer</b>												
Analyse des consignes	0:10	0:10										
<b>Planifier</b>												
Réalisation du planning	1:00	1:00										
Réalisation du backlog	1:30	1:30										
Mise en place d'un dépôt git	0:10	0:10										
Rédaction des scénarios de test	1:30	1:30										
<b>Réalisation du projet</b>												
S5 Installation Composer / routeur et autoloader	0:30	0:30										
S6 Mise en place d'un Virtual Host	0:15	0:15										
S2 Installation de la base de données	0:45	0:45										
S3 Création du style du site	5:20		3:00	2:20								
S4 Connexion à la base de données en PHP	0:10	0:10										
S7 Affichage des fiches sur la page d'accueil	1:00			1:00								
S9 Filtrer les fiches	1:30					1:30						
S8 Recherche de fiches sur la page d'accueil	1:30					1:30						
S14 Gestion des avis	3:00				3:00							
S10 Inscription	1:00		1:00									
S11 Connexion	1:00		1:00									
S13 Déconnexion	0:30					0:30						
S15 Suppression des avis	0:30					0:30						
S16 Page profil	3:00						3:00					
S17 Détails des fiches de jeux	1:00				1:00							
S12 Gestion des fiches de jeux admin	2:00			2:00								
Documentation	46:10	2:00	2:20	2:00	2:10	3:00	3:00	6:40	6:00	6:30	6:30	6:00
<b>Contrôles</b>												
Test des nouvelles fonctionnalités	2:00		0:20	0:20	0:30	0:30	0:20					
Test journaliers	4:40		0:20	0:20	0:20	0:30	0:20	0:20	0:30	0:30	0:30	1:00
Correction de bugs	7:50				1:00		1:20	1:00	1:30	1:00	1:00	1:00
	88:00	8:00	8:00	8:00	8:00	8:00	8:00	8:00	8:00	8:00	8:00	8:00

## Planning réel

Le planning effectif, est le planning du suivi du projet. Il permet de voir les différences avec le planning prévisionnel.

Tâches à réaliser	Temps nécessaire	27. avr 2023	2. mai 2023	3. mai 2023	4. mai 2023	8. mai 2023	9. mai 2023	10. mai 2023	11. mai 2023	15. mai 2023	16. mai 2023	17. mai 2023
<b>S'informer</b>	<b>0:10</b>											
Analyse des consignes	0:10	0:10										
<b>Planifier</b>	<b>4:05</b>											
Réalisation du planning	1:25	1:25										
Réalisation du backlog	1:30	1:30										
Mise en place d'un dépôt git	0:10	0:10										
Rédaction des scénarios de test	1:00		1:00									
<b>Réalisation du projet</b>	<b>65:50</b>											
<b>\$5</b> Installation Composer / routeur et autoloader	0:30	0:30										
<b>\$6</b> Mise en place d'un Virtual Host	0:15	0:15										
<b>\$2</b> Installation de la base de données	0:50	0:50										
<b>\$3</b> Création du style du site	11:05	1:00	1:00	1:20	2:30	2:10	2:15	0:50				
<b>\$4</b> Connexion à la base de données en PHP	0:10	0:10										
<b>\$7</b> Affichage des fiches sur la page d'accueil	1:44		0:50		0:54							
<b>\$9</b> Filtrer les fiches	0:41				0:11	0:30						
<b>\$8</b> Recherche de fiches	0:23				0:23							
<b>\$14</b> Gestion des avis	2:20			1:30	0:50							
<b>\$10</b> Inscription	0:45		0:45									
<b>\$11</b> Connexion	0:20		0:20									
<b>\$13</b> Déconnexion	0:25		0:25									
<b>\$15</b> Suppression des avis	0:30			0:30								
<b>\$16</b> Page profile	1:20					1:20						
<b>\$17</b> Détails des fiches de jeux	1:00			1:00								
<b>\$12</b> Gestion des fiches de jeux admin	3:00		1:40	0:30	0:50							
<b>\$18</b> Mot de passe oublié	1:20					0:40	0:40					
<b>\$19</b> Pagination page jeux	1:00							1:00				
<b>\$20</b> Pagination page gestion	3:15								3:15			
Documentation	34:57	2:00	1:20	2:00	1:22	2:30	3:15	3:00	3:00	6:00	7:00	3:30
<b>Contrôles</b>	<b>13:55</b>											
Test des nouvelles fonctionnalités	1:50		0:20	0:20	0:10	0:10	0:20	0:10	0:20			
Test journaliers	5:25		0:20	0:20	0:10	0:10	0:30	1:00	0:25	1:00	1:00	0:30
Correction de bugs	6:40			0:30	0:40	0:30	1:00	2:00	1:00	1:00		
	<b>84:00</b>	8:00	8:00	8:00	8:00	8:00	8:00	8:00	8:00	8:00	8:00	4:00

Légende :

Tâches non prévues

# Analyse fonctionnelle

## Fonctionnalités

### Inscription

L'inscription d'un utilisateur se passe en plusieurs étapes. Quand l'utilisateur aura complété les champs obligatoires, une vérification va être faite pour savoir si les mots de passe sont correspondants. Si cette étape est passée, une 2<sup>ème</sup> vérification va être faite pour savoir si le nom est déjà pris par un autre compte. Si ce n'est pas le cas, alors l'utilisateur est créé et il est redirigé sur la page de connexion. Si une de ses étapes échoue, un message d'erreur s'affiche. Par défaut les comptes utilisateurs n'ont pas les droits administrateurs.

### Connexion

Concernant la connexion, il y a d'abord une vérification sur le nom puis sur le mot de passe. Si l'une des deux échoue alors un message d'erreur s'affiche. Sinon l'utilisateur est connecté et ses données sont stockées en session.

### Mot de passe oublié

Si l'utilisateur oublie son mot de passe, il a la possibilité de cliquer sur un lien qui lui permet de recevoir un nouveau mot de passe généré aléatoirement. Avant de cliquer il doit remplir le champ nom car sinon le mot de passe ne peut pas être attribué.

Voici le code qui permet de le faire avec les explications aux différentes étapes :

- Récupération des données du formulaire

```
var form = document.getElementById('form');  
var formData = new FormData(form);
```

- Création de la requête http, ouverture en précisant la méthode d'envoi (GET, POST) et l'url vers la page de traitement.

```
var xhr = new XMLHttpRequest();  
xhr.open('POST', '/mdpOublier');  
xhr.send(formData);
```

- Une fois les données envoyées, elles sont récupérées dans une fonction qui d'abord vérifie si le nom est rempli. Si c'est le cas, une nouvelle vérification est faite pour savoir si l'utilisateur existe. Si tout correspond alors un nouveau mot de passe est généré de

cette façon. Je crée un tableau qui contient toutes les lettres de l'alphabet en minuscule et en majuscule et les nombres de 0 à 9. Ensuite un tirage aléatoire se fait dans le tableau jusqu'à ce que le mot de passe fasse la taille demandée. (Par défaut la taille est à 8).

```
$characters = array_merge(range('a', 'z'), range('A', 'Z'), range(0, 9));  
$password = "";  
  
for ($i = 0; $i < $length; $i++)  
    $password .= $characters[rand(0, count($characters) - 1)];  
}  
return $password;
```

- Ensuite pour envoyer la réponse, il faut mettre un `json_encode()` avec le message en paramètre.
- Affichage de la réponse

```
xhr.addEventListener('load', function() {  
    if (xhr.status === 200) {  
        alert(xhr.responseText);  
    } else {  
        console.error('Erreur de traitement du serveur');  
    }  
});
```

## Création d'un jeu

Pour la création d'un jeu, il y a d'abord une vérification sur l'image puis sur les types pour savoir s'il y en a au moins un qui est sélectionné. Si tout est bon, alors le fichier est renommé avec un nom unique. Pour ce faire j'ai utilisé la fonction PHP « `uniqid()` » qui permet de générer une liste de nombres uniques. Ensuite l'image est déplacée dans un dossier qui est dans le projet en utilisant « `move_uploaded_file()` ». Ensuite le jeu se crée et l'identifiant du jeu est récupéré pour attribuer les différents types au jeu.

```
$fileName = GamesModel::NameFileRandom($fileName);  
  
move_uploaded_file($file_tmp, "assets/images/" . $fileName);  
  
$idGame = GamesModel::CreateGame($date, $description, $title, $fileName, $platform);  
  
// Attribue les types au jeu  
foreach ($types as $key => $type) {  
    $idType = TypesModel::GetIdTypeByName($type);  
    TypesModel::GiveTypeToGame($idGame, $idType->id); }  
}
```

## Modification d'un jeu



Lors de la modification d'un jeu, si aucun type n'est coché, alors un message d'erreur s'affiche sinon il y a une vérification sur l'image. Si aucune image n'est sélectionnée, automatiquement le jeu gardera l'image précédente sinon elle se modifie.

Ensuite pour éviter les répétitions, si un type était déjà attribué à ce jeu alors il ne se passe rien sinon il est ajouté et pour finir les types qui étaient déjà attribués et qui ne sont pas gardés sont supprimés. Voici le code correspondant à l'explication :

```
$typesOfGame = TypesModel::GetTypesOfGame($idGame);
$typeAlreadyExist = false;

// Attribution des types qui ne sont pas encore dans la base de données
foreach ($newTypes as $key => $newType) {
    foreach ($typesOfGame as $key => $typeExist) {
        if ($typeExist->type == $newType) {
            $typeAlreadyExist = true;
        }
    }
    if (!$typeAlreadyExist) {
        $idType = TypesModel::GetIdTypeByName($newType);
        TypesModel::GiveTypeToGame($idGame, $idType->id);
    }
    $typeAlreadyExist = false;
}

// Suppression des types qui n'ont pas été cochés
$typesToDelete = array();
foreach ($typesOfGame as $key => $typeExist) {
    if (!in_array($typeExist->type, $newTypes)) {
        array_push($typesToDelete, TypesModel::GetIdTypeByName($typeExist->type));
    }
}
foreach ($typesToDelete as $typeId) {
    TypesModel::DeleteTypeOfGame($idGame, $typeId->id);
}
```

## Suppression d'un jeu

La suppression d'un jeu récupère seulement l'identifiant du jeu dans l'url puis supprime le jeu qui a cet identifiant.

### Création d'un commentaire

Pour créer un commentaire, il faut être connecté puis il faut remplir au moins les champs nom et note puis le commentaire s'affiche directement en dessous du jeu.

### Suppression d'un commentaire

Une fois le bouton cliquer, comme pour le jeu, le commentaire récupère seulement l'identifiant du commentaire puis le supprime.

### Modification d'un commentaire

La modification de commentaire est similaire à la création.

### Modification du profil

La modification du profil comprend à la fois la modification du mot de passe et de l'avatar. L'utilisateur qui veut modifier son mot de passe doit rentrer son ancien mot de passe, le nouveau ainsi qu'une confirmation. Il a une case qui est cochée par défaut qui permet de garder l'avatar précédent. Si cette case n'est pas cochée et qu'aucun avatar n'est sélectionné, alors l'utilisateur aura l'avatar par défaut sinon il prendra le nouveau.

### Affichage des jeux commentés dans le profil

Pour afficher les jeux que l'utilisateur a commenté, il faut récupérer la liste de tous les identifiants des jeux qui correspondent à l'identifiant de l'utilisateur.

```
SELECT DISTINCT jeux.id FROM jeux INNER JOIN avis ON jeux.id = avis.jeux_id INNER JOIN utilisateurs ON utilisateurs.id = avis.utilisateurs_id AND utilisateurs.id = ?
```

La seule différence avec une requête classique, c'est qu'il y a le mot « DISTINCT » qui permet de ne pas avoir plusieurs fois le même identifiant. Cela me permet de n'afficher qu'une seule fois un jeu que l'utilisateur a commenté plus d'une fois.

## Recherche

La recherche s'effectue sur le titre et la description. Il suffit de récupérer la recherche et de la passer dans cette fonction :

```
public static function SearchGamesByTitleOrDescription($search)
{
    $sql = "SELECT * FROM jeux WHERE description LIKE ? OR titre LIKE ? ";

    $param = ["%$search%", "%$search%"];
    return ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);
}
```

## Filtrages

Il y a deux types de filtrages, par date et par note.

Le filtrage par date tri les jeux en fonction de leurs dates et de leurs identifiants, car si plusieurs jeux ont été créés le même jour alors on ne pourra pas savoir lequel est le plus récent. Voilà pourquoi je fais une vérification sur les identifiants en plus de la date.

Le filtrage par note tri les jeux en fonction de la moyenne de leurs notes. Pour ce faire j'ai créé plusieurs fonctions que je vais un peu détailler.

```
public static function AverageMarks($marks)
{
    $average = 0;
    $nbMarks = count($marks);

    foreach ($marks as $key => $mark) {
        $average += $mark->note;
    }

    if ($nbMarks != 0) {
        $average = (double)($average / $nbMarks);
    } else {
        $average = 0;
    }

    return $average;
}
```

Cette première fonction permet de calculer la moyenne des notes. Si il y a au moins une note alors on les additionne puis on la divise par le nombre de notes qu'on retourne en « double ».

La fonction précédente est appelée dans celle-ci :

Cette fonction récupère les moyennes puis les tris dans l'ordre de la plus grande à la plus petite. Elle va au final retourner un tableau avec l'identifiant des jeux ordonnés.

```
public static function SortedGamesByAverageMark()  
{  
    $games = self::GetAllGames();  
  
    $averageMarks = array();  
    foreach ($games as $game) {  
        $marks = self::GetMarksOfGame($game->id);  
        if (count($marks) > 0) {  
            $average = self::AverageMarks($marks);  
            $averageMarks[$game->id] = $average;  
        } else {  
            $averageMarks[$game->id] = 0;  
        }  
    }  
  
    // Tri décroissant des moyennes  
    arsort($averageMarks);  
  
    $sortedGames = array();  
    foreach ($averageMarks as $gameId => $averageMark) {  
        $sortedGames[] = self::GetGameById($gameId);  
    }  
  
    return $sortedGames;  
}
```

## Pagination

Pour la pagination, il y a une fonction qui permet d'afficher la pagination en rapport avec le nombre total de jeux à afficher ainsi que le nombre de jeux par page.

```
public static function DisplayPagination($games, $limitGameToPage, $pageName){
    $output = " <div class='row text-start pt-5 border-top paginationCenter'>
        <div class='col-md-12'>
            <div class='custom-pagination'>
                <a href='$pageName?page=1'>1</a>";
    $page = 1;
    for ($i=1; $i <= count($games); $i++) {
        if ($i == $limitGameToPage * $page) {
            $page++;
            $output .= "<a href='$pageName?page=$page'>$page</a>";
        }
    }
    $output .= "</div></div></div>";
    return $output;
}
```

Ensuite cette fonction permet de récupérer le numéro de la ligne par laquelle il faut commencer à récupérer les jeux dans la base de données.

```
public static function GetOffset($page, $limitGameToPage){
    if (isset($page) && $page != 1) {
        $offset = $limitGameToPage * ($page - 1);
    }
    else{
        $page = 1;
        $offset = $limitGameToPage * ($page - 1);
    }
    return $offset;
}
```

Pour pouvoir passer tous les jeux qui doivent être affichés dans la requête, je retourne un « implode » qui permet de retourner le tableau sous la forme d'un string séparé par des virgules.

```
public static function GetStrIdsGames($games){
    $idGames = array();
    foreach ($games as $game) {
        array_push($idGames, $game->id);
    }
    return implode(',', $idGames); }
```

Pour finir une fois que le tableau est retourné sous la forme d'un « string », j'ai utilisé « IN » qui me permet de chercher les jeux dans la liste d'identifiants, puis de les retourner dans le même ordre qu'ils ont été passés en utilisant « ORDER BY FIELD(id, \$idGames) ». Cette ligne fait, que les champs vont être retournés dans l'ordre des identifiants par rapport au « string » « \$idGames ».

```
public static function GetGamesByLimit($offset, $next, $idGames){  
    $sql = "SELECT * FROM jeux WHERE id IN ($idGames) ORDER BY FIELD(id, $idGames)  
LIMIT ?, ?";  
  
    $param = [$offset, $next];  
    return ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);  
}
```

Elle va finalement, retourner tous les jeux correspondant pour pouvoir les afficher.

# Analyse organique

## Structure du projet

La structure de l'application a été réalisée en MVC (modèle, vue contrôleur). Chaque fichier est classé dans le dossier correspondant.

public/.htaccess

Le fichier qui permet la réécriture d'url et de tout rediriger vers l'index.php

public/index.php

Le fichier principal qui démarre le routeur et redirige sur routes.php.

public/assets/

Le dossier contient toute la partie médias, design. Il contient toute la css, tout le javascript ainsi que toutes les images de l'application.

routes/routes.php

Le fichier qui contient toutes les routes et qui redirige en fonction de la page à afficher.

src/Views/

Le dossier qui contient toutes les vues.

src/Models/

Le dossier qui contient toutes les fonctions et la logique liées à la base de données.

src/Controllers/

Le dossier qui contient un contrôleur par vue qui permet de faire le lien entre le modèle et la vue.

src/vendor/

Le dossier contient toutes les librairies installées. C'est le dossier généré par Composer.

src/Views/inc

Le dossier contient le « header » et le « footer » du site.

sql/

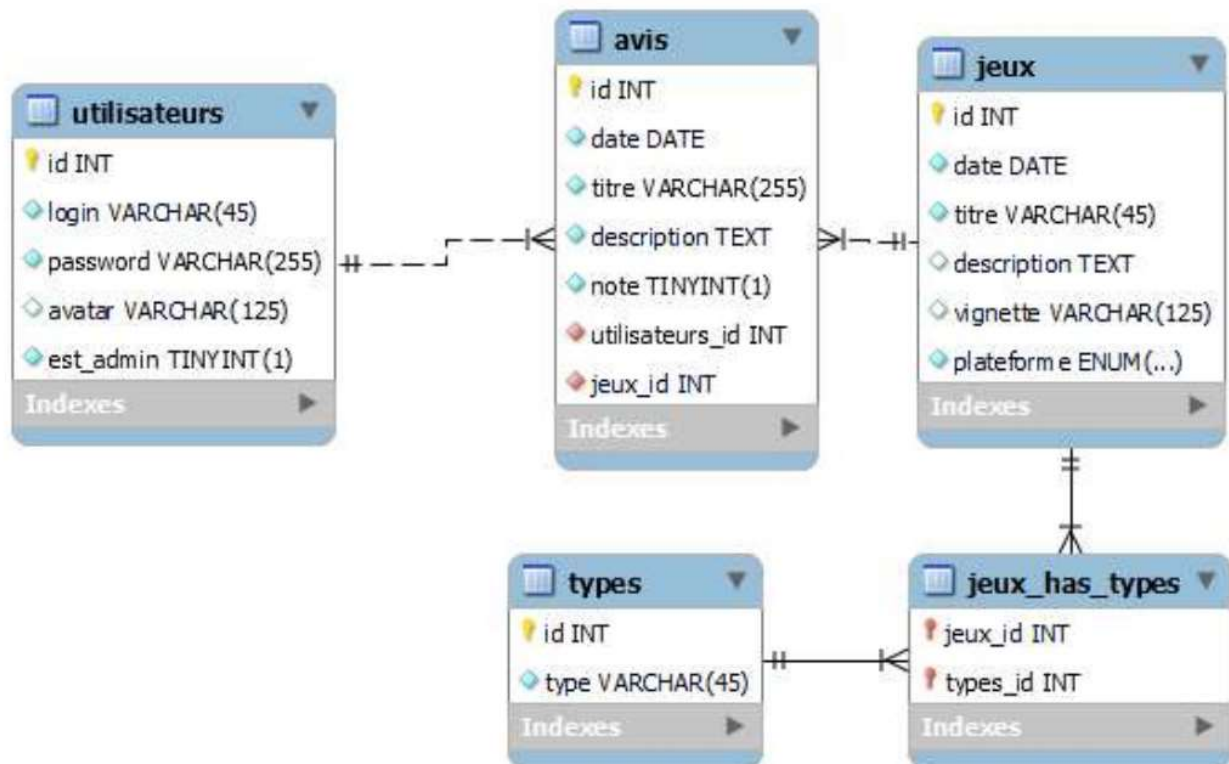
Le dossier contient l'exportation de la base de données.

composer.json

Fichier de configuration qui permet de spécifier les dépendances requises pour le projet.

```
> .vscode
  > public
    > assets
    .htaccess
    index.php
  > routes
    routes.php
  > sql
    db_shareGames.sql
  > src
    > Controllers
      CreateGameController.php
      DeleteController.php
      DetailsGameController.php
      FunctionsController.php
      GamesController.php
      HomeController.php
      LoginController.php
      LogoutController.php
      ManagementGamesController.php
      ProfileController.php
      RegisterController.php
      UpdateController.php
    > Models
      ConnexionDB.php
      GamesModel.php
      OpinionsModel.php
      TypesModel.php
      UsersModel.php
    > Views
      > inc
        footer.php
        header.php
      createGameView.php
      detailsGameView.php
      gamesView.php
      homeView.php
      loginView.php
      managementGamesView.php
      profileView.php
      registerView.php
      updateGameView.php
      updateOpinionView.php
      updateProfileView.php
    config_exemple.php
    config.php
  > vendor
    {} composer.json
    {} composer.lock
```

## Base de données





# Librairies et outils externes

## Bootstrap

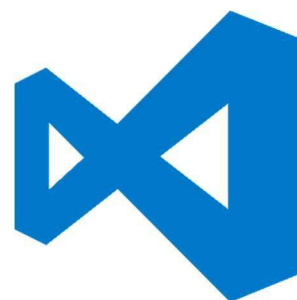
Bootstrap est une puissante librairie de développement front-end utilisée pour créer des interfaces modernes et réactives et flexibles. Elle fournit un ensemble complet de composants, de styles prédéfinis et de fonctionnalités JavaScript pour accélérer le processus de conception d'un site web.



## Visual Studio Code

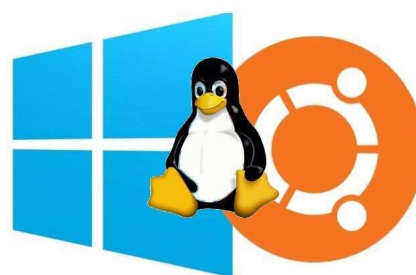
Visual studio code est un environnement de développement facile à utiliser et complet. Il permet de coder dans différents langages et beaucoup d'extensions sont disponibles.

Je l'ai utilisé pour développer mon site web.



## WSL 2 (Ubuntu)

WSL 2 est un sous-système Linux dans Windows. Il permet en quelque sorte d'avoir une machine virtuelle Linux tout en étant relié à la machine Windows. J'ai utilisé ubuntu 20.04 pour me connecter.



## Peccee/simple-router

Peccee simple-router est une librairie fortement inspirée de la façon dont Laravel gère ses routages. Il est rapide et puissant et il est facile de l'intégrer dans un projet. C'est un outil utile pour gérer le routage dans une application web PHP et il offre une manière simple et flexible de définir les routes dans une application.

## Github

Github est un outil de versionnage GitHub est une plate-forme d'hébergement de code pour le contrôle de version et la collaboration. Il permet de travailler à plusieurs sur des projets et d'avoir un historique de versions.



## Composer

Composer est un logiciel de gestionnaire de dépendances libre écrit PHP. Il permet d'installer des bibliothèques dans un projet. C'est grâce à composer que j'ai pu inclure Pecee/simple-router et autoload.



en

## Autoload

L'autoload est un mécanisme qui permet de charger automatiquement les classes PHP sans avoir à les inclure manuellement avec la fonction «include» ou «require». Il sert aussi à définir un namespace et choisir le dossier cible qui contient les classes. En résumé, il facilite la gestion des classes dans les applications PHP et réduit les erreurs liées aux inclusions manuelles de fichiers.

## Périmètre de test

J'ai réalisé mes tests avec Google Chrome, Microsoft Edge et Firefox. Pour effectuer les tests j'ai utilisé un compte administrateur, un compte lambda et aussi en tant que visiteur. J'ai fait un plan de test que j'ai utilisé tous les jours pour vérifier qu'une nouvelle fonctionnalité ne crée pas de bugs dans une autre.

## Équipement de test

Environnement :

- Google Chrome 113.0.5672.63 (64 bits) sur Windows 10
- Microsoft Edge 113.0.1774.35 (64 bits) sur Windows 10
- Firefox 106.0.5 (64 bits) sur Windows 10

Pour le serveur, j'ai utilisé un serveur Apache version 2.4.41 qui fonctionne avec Ubuntu. Concernant le PHP j'utilise la version 8.1 ainsi qu'un serveur maria dB version 10.3.34

# Plan de test

Un plan de test des principales fonctionnalités de l'application est disponible ci-dessous.

Nom	1.1 <b>Routage</b>
User story	S5 Installation et configuration de composer et des librairies externes
Situation	Quand on arrive sur le site une page d'accueil s'affiche.
Résultat attendu	Le système de routage est fonctionnel et les pages s'affichent en fonction de leur url.

Nom	2.1 <b>Affichage des fiches de jeux sur l'accueil</b>
User story	S7 Affichage des jeux sur la page d'accueil
Situation	Quand on arrive sur la page d'accueil, la liste des 10 meilleurs jeux est affichée ainsi que les 5 jeux les plus récents.
Résultat attendu	Les bonnes fiches de jeux s'affichent.

Nom	<b>3.1 Recherche de jeux</b>
User story	S8 Recherche de jeux
Situation	Quand on recherche un jeu, les jeux contenant la recherche dans le titre ou la description sont affichés
Résultat attendu	Les jeux s'affichent selon la recherche.

Nom	<b>4.1 Filtrage des jeux par date</b>
User story	S9 Affichage et filtrage des jeux sur une page
Situation	Quand on choisit le filtrage par date, les fiches de jeux sont affichées par ordre chronologique selon la date de création / modification.
Résultat attendu	Les fiches de jeux sont affichées par ordre chronologique.

Nom	<b>4.2 Filtrage des jeux par note</b>
User story	S9 Affichage et filtrage des jeux sur une page
Situation	Quand on choisit le filtrage par note, les fiches de jeux sont affichées par ordre croissant selon la moyenne des notes.
Résultat attendu	Les fiches de jeux sont affichées par ordre croissant.

Nom	<b>5.1 Création d'un compte</b>
User story	S10 Inscription
Situation	Quand on valide l'inscription, si le nom d'utilisateur existe déjà une erreur s'affiche sinon le compte se crée et on est redirigé sur la page de connexion.
Résultat attendu	Une erreur s'affiche ou le compte se crée.

Nom	6.1 <b>Connexion à un compte</b>
User story	S11 Connexion
Situation	Quand on se connecte, un message d'erreur s'affiche si les informations sont incorrectes. Sinon, l'utilisateur est connecté.
Résultat attendu	Une erreur s'affiche ou l'utilisateur est connecté.

Nom	7.1 <b>Création d'une fiche de jeu</b>
User story	S12 Gestion des fiches de jeux
Situation	Quand on clique sur le bouton de création.
Résultat attendu	La fiche de jeu s'affiche dans les différentes pages.

Nom	7.2 <b>Suppression de fiches de jeux</b>
User story	S12 Gestion des fiches de jeux
Situation	Quand on clique sur le bouton de suppression.
Résultat attendu	La fiche de jeu est supprimée.

Nom	7.3 <b>Modification de fiches de jeux</b>
User story	S12 Gestion des fiches de jeux
Situation	Une fois le formulaire rempli lorsqu'on clique sur modifier.
Résultat attendu	La fiche de jeu est modifiée avec les nouvelles informations ou un message s'affiche en fonction de l'erreur.

Nom	8.1 <b>Déconnexion</b>
User story	S13 Déconnexion
Situation	Lorsqu'on souhaite se déconnecter, on a un lien déconnexion qui le permet.
Résultat attendu	Déconnexion du compte.

Nom	9.1 <b>Création d'un avis</b>
User story	S14 Gestion des avis
Situation	Une fois les champs du formulaire rempli, lorsqu'on clique sur le bouton.
Résultat attendu	Le commentaire est créé ou un message s'affiche correspondant à l'erreur.

Nom	9.2 <b>Modification d'un avis</b>
User story	S14 Gestion des avis
Situation	Quand on valide la modification, si les champs sont remplis et corrects alors le commentaire se modifie sinon un message d'erreur s'affiche.
Résultat attendu	Le commentaire est modifié ou un message s'affiche.

Nom	9.3 <b>Suppression d'un avis</b>
User story	S14 Gestion des avis
Situation	Quand on clique sur le bouton de suppression.
Résultat attendu	L'avis est supprimé et n'apparaît plus dans les différentes pages.

Nom	10.1 <b>Suppression des avis (Administrateur)</b>
User story	S15 Suppression des avis
Situation	Quand on clique sur le bouton de suppression.
Résultat attendu	L'avis est supprimé et n'apparaît plus dans les différentes pages.

Nom	11.1 <b>Modification mot de passe</b>
User story	S16 Page profile
Situation	Quand on clique sur le bouton de suppression.
Résultat attendu	Le mot de passe est modifié.

Nom	11.2 <b>Modification avatar</b>
User story	S16 Page profile
Situation	Une fois les champs modifiés si on clique sur le bouton de modification.
Résultat attendu	L'avatar se modifie ou un message d'erreur s'affiche.

Nom	11.3 <b>Affichage des fiches de jeux commentées</b>
User story	S16 Page profile
Situation	Quand on arrive sur la page profile, il y a la liste des fiches de jeux ou on a posté un ou plusieurs commentaires.
Résultat attendu	La liste s'affiche.



Nom	12.1 <b>Affichage d'une fiche de jeu</b>
User story	S17 Détails des fiches de jeux
Situation	Quand on arrive sur la page de détails d'une fiche de jeu.
Résultat attendu	Les commentaires s'affichent dans l'ordre chronologique sous la fiche.

Nom	13.1 <b>Oublie du mot de passe</b>
User story	S18 Mot de passe oublié
Situation	Quand on est sur la page de connexion, si on a oublié notre mot de passe il y a un lien qui permet de générer un mot de passe aléatoire.
Résultat attendu	une pop-pop s'affiche avec le nouveau mot de passe

Nom	14.1 <b>Pagination jeux</b>
User story	S19 Pagination page jeux
Situation	Quand on est sur la page de jeux on doit pouvoir naviguer entre différentes pages afin de voir tous les jeux à afficher.
Résultat attendu	Un certain nombre de jeux s'affichent par page et une pagination s'affiche par rapport au nombre de jeux à afficher. S'il y a un filtre ou une recherche, ils doivent rester actifs entre les différentes pages.

Nom	14.2 <b>Pagination gestion</b>
User story	S20 Pagination page gestion
Situation	Quand on est sur la page de gestion on doit pouvoir naviguer entre différentes pages afin de voir tous les jeux.
Résultat attendu	Un certain nombre de jeux s'affichent et une pagination s'affiche par rapport au nombre de jeux à afficher.

## Évolution des tests

N. test	J1 jeu. 27.04	J2 mar. 02.05	J3 mer. 03.05	J4 je. 04.05	J5 lun. 08.05	J6 mar. 09.05	J7 mer. 10.05	J8 jeu. 11.05	J9 lun. 15.05	J10 mar. 16.05	J11 mer. 17.05
1.1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2.1	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3.1	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
4.1	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
4.2	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
5.1	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
6.1	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7.1	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7.2	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
7.3	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
8.1	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
9.1	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
9.2	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
9.3	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
10.1	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
11.1	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
11.2	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
11.3	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
12.1	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
13.1	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
14.1	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
14.2	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓

# Rapport de test

Nom	1.1 <b>Routage</b>
Date	27.04.2023
Résultat obtenu	La page d'accueil c'est affiché.
Statut	OK

Nom	5.1 <b>Création d'un compte</b>
Date	02.05.2023
Résultat obtenu	Le compte a été créé.
Statut	OK

Nom	6.1 <b>Connexion à un compte</b>
Date	02.05.2023
Résultat obtenu	La connexion a été obtenue sinon un message d'erreur s'affiche.
Statut	OK

Nom	7.1 <b>Création d'une fiche de jeu</b>
Date	02.05.2023
Résultat obtenu	Erreur sur la date. Mauvais format.
Statut	KO

Nom	7.1 <b>Création d'une fiche de jeu</b>
Date	02.05.2023
Résultat obtenu	Fiche de jeu créé avec succès aucune erreur sauf en cas d'image non sélectionnée.
Statut	OK

Nom	2.1 <b>Affichage des fiches de jeux sur l'accueil</b>
Date	02.05.2023
Résultat obtenu	Les fiches s'affichent mais l'image ne fonctionne pas
Statut	KO

Nom	2.1 <b>Affichage des fiches de jeux sur l'accueil</b>
Date	02.05.2023
Résultat obtenu	Les fiches s'affichent.
Statut	KO

Nom	9.1 <b>Création d'un avis</b>
Date	03.05.2023
Résultat obtenu	L'avis s'est créé
Statut	OK

Nom	12.1 <b>Affichage d'une fiche de jeu</b>
Date	03.05.2023
Résultat obtenu	Le jeu s'affiche avec les commentaires en dessous trié dans l'ordre chronologique
Statut	OK

Nom	10.1 <b>Suppression des avis (Administrateur)</b>
Date	03.05.2023
Résultat obtenu	Le commentaire a été supprimé et ne s'affiche plus.
Statut	OK

Nom	9.3 <b>Suppression d'un avis</b>
Date	03.05.2023
Résultat obtenu	Le commentaire a été supprimé et ne s'affiche plus.
Statut	OK

Nom	7.2 <b>Suppression de fiches de jeux</b>
Date	03.05.2023
Résultat obtenu	La fiche de jeux a été supprimé et les commentaires qui sont liés sont aussi supprimés.
Statut	OK

Nom	7.3 <b>Modification de fiches de jeux</b>
Date	04.05.2023
Résultat obtenu	La modification marche mais le titre et la description s'échangent.
Statut	KO

Nom	2.1 <b>Affichage des fiches de jeux sur l'accueil</b>
Date	04.05.2023
Résultat obtenu	Les 10 meilleurs jeux s'affichent en fonction de leur moyenne
Statut	OK

Nom	9.2 <b>Modification d'un avis</b>
Date	04.05.2023
Résultat obtenu	L'avis se modifie
Statut	OK

Nom	7.3 <b>Modification d'une fiche de jeux</b>
Date	04.05.2023
Résultat obtenu	La fiche de jeux se modifie correctement et s'il n'y pas d'image elle garde la précédente. Elle affiche une erreur si aucun type n'est sélectionné.
Statut	OK

Nom	3.1 <b>Recherche de jeux</b>
Date	04.05.2023
Résultat obtenu	Les jeux qui ont la recherche dans la description ou le titre sont affichés.
Statut	OK

Nom	4.1 <b>Filtrage des jeux par date</b>
Date	04.05.2023
Résultat obtenu	Les jeux se sont affichés du plus récent au plus ancien.
Statut	OK

Nom	11.3 <b>Affichage des fiches de jeux commentées</b>
Date	08.05.2023
Résultat obtenu	Les jeux commentés par l'utilisateur s'affichent mais avec des doublons s'il y a plusieurs commentaires dans un jeu.
Statut	KO

Nom	11.3 <b>Affichage des fiches de jeux commentées</b>
Date	08.05.2023
Résultat obtenu	Les jeux commentés par l'utilisateur s'affichent correctement et sans doublons.
Statut	OK

Nom	11.1 <b>Modification mot de passe</b>
Date	08.05.2023
Résultat obtenu	Le mot de passe se modifie.
Statut	OK

Nom	11.1 <b>Modification mot de passe</b>
Date	08.05.2023
Résultat obtenu	Si l'ancien mot de passe ou les nouveaux ne sont pas identiques une erreur s'affiche.
Statut	OK

Nom	11.2 <b>Modification avatar</b>
Date	08.05.2023
Résultat obtenu	L'avatar se modifie si on le change. Si aucun avatar est sélectionné mais que la case « garder mon avatar » est cochée alors l'avatar ne change pas sinon il est effacé.
Statut	OK

Nom	4.2 <b>Filtrage de jeux par note</b>
Date	08.05.2023
Résultat obtenu	Les jeux se sont affichés de la meilleure moyenne de notes à la moins bonne ou non noté.
Statut	OK

Nom	13.1 <b>Mot de passe oublié</b>
Date	09.05.2023
Résultat obtenu	Le nouveau mot de passe s'affiche dans le site.
Statut	OK



Nom	14.2 <b>Pagination gestion</b>
Date	10.05.2023
Résultat obtenu	Les différents jeux s'affichent avec le nombre de jeux définis. La pagination se crée automatiquement en fonction du nombre de jeux qui doivent être affichés
Statut	OK

Nom	14.1 <b>Pagination jeux</b>
Date	11.05.2023
Résultat obtenu	Les différents jeux s'affichent dans les pages et la pagination fonctionne. S'il y a un filtrage ou une recherche, lors d'un changement de page le filtre ou la recherche se réinitialise.
Statut	KO

Nom	14.1 <b>Pagination jeux</b>
Date	11.05.2023
Résultat obtenu	Les différents jeux s'affichent dans les pages, la pagination fonctionne et le filtrage est gardé pendant la navigation entre les pages.
Statut	OK

# Conclusion

## Planification

Contrairement à mon planning prévisionnel, j'ai avancé un petit peu plus vite que ce que j'avais prévu. Malgré cela, j'ai aussi remarqué que mon estimation de temps sur certaines tâches était soit trop élevé soit trop faible. Notamment le CSS qui devait me prendre environ 5 heures et qui m'en a finalement pris 11. Pour le reste je n'ai jamais pris de retard et j'ai pu faire des tâches bonus qui permettent une meilleure ergonomie sur le site.

## Améliorations possibles

Concernant les améliorations possibles si le projet est repris, il y aurait :

- Modifier la recherche afin qu'elle s'effectue en même temps que l'utilisateur écrit (AJAX).
- Ajouter un mail à l'utilisateur afin d'utiliser un serveur SMTP (Simple Mail Transfert Protocol), et donc envoyer un email lors de l'oubli de son mot de passe.
- Pouvoir filtrer les jeux par types.

Le projet est fonctionnel et toutes les fonctionnalités demandées ont été implémentées.

## Bilan personnel

J'ai plutôt bien aimé effectuer ce travail, car, tout d'abord, c'est le travail qui peut me permettre d'obtenir mon CFC. Ensuite le projet est centré sur les jeux vidéo qui est un sujet qui m'intéresse beaucoup. J'ai ressenti une légère baisse d'enthousiasme lorsqu'il s'agissait de rédiger la documentation, car je la trouvais assez longue et selon moi c'est un domaine qui n'a jamais été réellement entraînée au sein de la formation. Cependant, comme j'ai décidé de l'avancer simultanément au projet, une fois qu'il était terminé il ne me restait plus grand-chose à faire dans la documentation.

## Remerciements

Je tiens à remercier :

- Monsieur Garchery pour son accompagnement durant mon TPI
- Monsieur Malambu pour son accompagnement durant mes 4 ans et surtout la 1<sup>ère</sup> année

# ANNEXES

## Bibliographie

### Template

<https://themewagon.com/themes/blogy/>

### Sites web utilisés

<https://sql.sh/>

[www.w3schools.com](http://www.w3schools.com)

<https://php.net/>

<https://www.javatpoint.com/how-to-change-date-format-in-php>

<https://fr.wikipedia.org/>

<https://chat.openai.com/>

J'ai utilisé ChatGPT afin qu'il me fasse des commentaires réalistes sur les différents jeux

Conversation :

fais moi un commentaire sur clash royale et mets une note sur 10 au jeu

ChatGPT

Clash Royale est un jeu de stratégie en temps réel qui a captivé des millions de joueurs à travers le monde. Avec ses batailles rapides et compétitives, il offre une expérience de jeu addictive et palpitante.

Le concept de Clash Royale est simple mais ingénieux. Vous devez construire un deck de cartes en utilisant une variété de troupes, sorts et structures défensives, puis les déployer stratégiquement sur le champ de bataille pour détruire les tours de votre adversaire tout en défendant les vôtres. La clé de la victoire réside dans la bonne gestion des ressources, la synchronisation des attaques et la prise de décisions tactiques rapides.

Le jeu offre une grande variété de cartes, chacune ayant ses propres caractéristiques et pouvoirs uniques, ce qui permet aux joueurs de personnaliser leur stratégie en fonction de leur style de jeu préféré. De plus, Clash

Royale propose un système de progression où vous pouvez améliorer vos cartes et débloquer de nouvelles, ajoutant ainsi une dimension supplémentaire à la profondeur stratégique du jeu.

Les graphismes colorés et les animations fluides de Clash Royale contribuent à son attrait visuel, tandis que les effets sonores et la musique immersive ajoutent à l'excitation des batailles. De plus, le jeu encourage l'interaction sociale grâce à la possibilité de rejoindre des clans, de participer à des événements en ligne et de défier des joueurs du monde entier.

En ce qui concerne ma note, je donnerais à Clash Royale un solide 8 sur 10. Le jeu offre une expérience de jeu addictive et compétitive, avec une grande variété de cartes et une communauté active. Cependant, certains joueurs pourraient être frustrés par le système de progression qui peut sembler un peu lent sans dépenser de l'argent réel. Néanmoins, si vous êtes prêt à investir du temps ou de l'argent, Clash Royale peut vous offrir des heures de plaisir stratégique.

j'aimerais que tu fasse un commentaire de 3 ligne et quil y est un titre

ChatGPT

Titre: Clash Royale : Un jeu de stratégie captivant

Commentaire: Clash Royale est un jeu de stratégie en temps réel qui captive les joueurs grâce à ses batailles rapides et compétitives. Avec ses cartes uniques et son gameplay addictif, il offre une expérience de jeu palpitante qui ne manquera pas de vous tenir en haleine. Préparez-vous à affronter des adversaires du monde entier dans ce titre captivant.

fais la meme chose pour fifa 23

ChatGPT

Titre: FIFA 23 : Une expérience de football immersive

Commentaire: FIFA 23 offre une expérience de football immersive qui plaira aux amateurs du sport le plus populaire au monde. Avec des graphismes améliorés, des mouvements de joueurs réalistes et une variété de modes de jeu, le jeu vous permet de plonger dans l'action sur le terrain. Affrontez vos adversaires en ligne, créez votre équipe ultime ou revivez des moments historiques dans ce titre captivant qui repousse les limites de la simulation de football.

et la note ?

ChatGPT

Pour FIFA 23, je donnerais une note de 8,5 sur 10. Le jeu offre une expérience de football immersive avec des graphismes impressionnants et des mouvements de joueurs réalistes. La variété des modes de jeu et la possibilité de créer votre équipe ultime ajoutent une profondeur et une rejouabilité supplémentaires. Cependant, certains joueurs pourraient être déçus par les microtransactions présentes dans le jeu. Néanmoins, pour les fans de football, FIFA 23 offre une expérience solide et divertissante.

fais toujours la meme chose pour black ops 2

ChatGPT

Titre: Call of Duty: Black Ops 2 - Une expérience de tir intense et palpitante

Commentaire: Black Ops 2 est un jeu de tir qui offre une expérience intense et palpitante aux amateurs d'action. Plongez au cœur d'un scénario captivant, mêlant guerre froide, conspirations et missions secrètes. Avec son mode multijoueur compétitif et son mode zombie addictif, Black Ops 2 offre une grande variété de gameplay et des heures de divertissement. Préparez-vous à l'action et à l'adrénaline avec ce titre emblématique de la franchise Call of Duty.

Note: 9 sur 10. Black Ops 2 est un jeu qui a su combiner avec succès une campagne solo captivante, un mode multijoueur compétitif et un mode zombie addictif. Son gameplay solide, ses graphismes immersifs et son scénario bien ficelé en font l'un des meilleurs épisodes de la série. Malgré son âge, il reste un incontournable pour les fans de jeux de tir.

fais la meme chose pour gta 5

ChatGPT

Titre: GTA 5 - Une immersion totale dans un monde ouvert époustouflant

Commentaire: GTA 5 offre une expérience de jeu sans pareil dans un monde ouvert dynamique et époustouflant. Plongez dans les rues animées de Los Santos, explorez un vaste territoire et vivez une histoire captivante remplie d'intrigues et de personnages mémorables. Avec son gameplay riche en possibilités, que ce soit dans les missions principales, les activités annexes ou le mode en ligne, GTA 5 offre une liberté totale et un divertissement sans fin. Préparez-vous à une aventure inoubliable dans l'un des meilleurs jeux de la franchise.

Note: 9,5 sur 10. GTA 5 est un chef-d'œuvre du monde ouvert, avec un niveau de détail impressionnant, une variété d'activités et une histoire riche. Son gameplay fluide, ses graphismes époustouflants et son monde vivant en font l'un des meilleurs jeux de sa génération. Bien que le mode en ligne puisse parfois souffrir de problèmes de stabilité, l'expérience globale offerte par GTA 5 reste inégalée. Un incontournable pour les fans d'action et d'aventure.

fais la meme chose pour uncharted 4

ChatGPT

Titre: Uncharted 4: A Thief's End - Une aventure épique et cinématographique

Commentaire: Uncharted 4: A Thief's End transporte les joueurs dans une aventure épique digne d'un film hollywoodien. Avec des graphismes à couper le souffle, une narration captivante et des séquences d'action palpitantes, le jeu offre une immersion totale dans l'univers de Nathan Drake. Les paysages exotiques, les énigmes stimulantes et les personnages charismatiques font de cette dernière itération de la série Uncharted une expérience inoubliable. Préparez-vous à vivre une aventure digne des plus grands blockbusters dans Uncharted 4.

Note: 9,5 sur 10. Uncharted 4: A Thief's End est un chef-d'œuvre du jeu d'aventure, offrant une combinaison parfaite de graphismes époustouflants, de gameplay fluide et d'une histoire captivante. Les performances des acteurs, les dialogues bien écrits et les séquences d'action spectaculaires font de ce jeu une expérience cinématographique immersive. Bien que certains pourraient souhaiter une plus grande variété d'énigmes, Uncharted 4 reste un incontournable pour les amateurs d'aventure et de narration de qualité.

fais la meme chose pour hogwarts legacy

ChatGPT

Titre: Hogwarts Legacy - Plongez dans une aventure magique extraordinaire

Commentaire: Hogwarts Legacy offre aux joueurs la possibilité de vivre une aventure magique extraordinaire dans l'univers emblématique de Harry Potter. En tant qu'étudiant à Poudlard, vous pourrez explorer les salles mystérieuses du château, apprendre des sorts puissants, interagir avec des personnages emblématiques et découvrir des secrets enfouis depuis longtemps. Avec des graphismes époustouflants et une immersion totale, Hogwarts Legacy promet une expérience de jeu fascinante et captivante pour les fans de la franchise. Préparez-vous à vous laisser ensorceler par ce titre magique.

Note: En attente de sortie. Hogwarts Legacy est un jeu très attendu et sa note finale dépendra de son contenu, de sa qualité de gameplay et de sa capacité à offrir une expérience fidèle à l'univers de Harry Potter. Les premières impressions sont prometteuses et les fans ont de grandes attentes pour ce jeu. Il faudra attendre sa sortie pour donner une évaluation définitive.

fais un commentaire négatif sur clash royale

ChatGPT

Commentaire négatif sur Clash Royale :

Bien que Clash Royale ait réussi à captiver de nombreux joueurs, il présente également certains aspects qui peuvent être décevants. Le système de progression du jeu peut parfois sembler lent, nécessitant soit un investissement considérable en temps, soit des dépenses d'argent réel pour progresser plus rapidement. Cela peut créer une sensation de déséquilibre entre les joueurs, favorisant ceux qui sont prêts à dépenser davantage. De plus, certaines stratégies dominantes peuvent rendre le jeu répétitif et prévisible, diminuant ainsi la diversité des expériences de jeu. En fin de compte, malgré son attrait initial, Clash Royale peut frustrer les joueurs qui recherchent une progression équitable et une variété de stratégies plus dynamiques.

Fais un titre et mets une note

ChatGPT

Titre: Clash Royale : Un jeu addictif mais avec des frustrations

Note: 7 sur 10

Commentaire: Clash Royale est un jeu qui a su captiver de nombreux joueurs grâce à son gameplay addictif et compétitif. Cependant, il présente certaines frustrations qui peuvent entacher l'expérience. Le système de

progression peut être lent et nécessiter des dépenses d'argent réel pour avancer rapidement, créant ainsi un déséquilibre entre les joueurs. De plus, certaines stratégies dominantes peuvent rendre le jeu répétitif. Malgré ces aspects, Clash Royale reste un divertissement captivant pour ceux qui sont prêts à investir du temps ou de l'argent, mais il peut ne pas convenir à ceux qui recherchent une expérience équilibrée et variée sans dépenses supplémentaires.

fais un commentaire avec titre et note sur god of war

ChatGPT

Titre: God of War : Une épopée épique et immersive

Note: 9,5 sur 10

Commentaire: God of War est un chef-d'œuvre qui plonge les joueurs dans une épopée épique et immersive. Avec des graphismes à couper le souffle, une histoire captivante et des combats d'une intensité incroyable, le jeu repousse les limites de l'action-aventure. Le nouvel aspect narratif et émotionnel ajouté à la série offre une profondeur inégalée, tandis que les énigmes stimulantes et les rencontres épiques avec des dieux et des créatures mythologiques rendent chaque instant mémorable. God of War est une véritable démonstration de l'excellence du jeu vidéo et un incontournable pour tous les amateurs de jeux d'action-aventure.

fais un commentaire négatif sur red evil 4

ChatGPT

Titre: Resident Evil 4 : Un pas en arrière dans la franchise

Note: 5 sur 10

Commentaire: Malheureusement, Resident Evil 4 marque un pas en arrière dans la franchise qui avait réussi à captiver les joueurs avec ses précédents opus. Bien que le jeu ait introduit des éléments de gameplay innovants à l'époque, tels que la caméra par-dessus l'épaule, il a également abandonné certains aspects clés qui ont fait le charme des jeux précédents. L'atmosphère effrayante et oppressante qui caractérisait les premiers jeux de la série a été diluée au profit d'une approche plus axée sur l'action. De plus, le scénario et les personnages ne parviennent pas à captiver autant que dans les titres antérieurs. Resident Evil 4 peut satisfaire les fans de jeux d'action, mais il déçoit ceux qui recherchent l'expérience classique de survival-horror offerte par les premiers jeux de la franchise.

fais un commentaire titre et note sur black ops 3 puis fais sur black ops 4

ChatGPT

Titre: Black Ops 3 : Une expérience de tir futuriste intense

Note: 8 sur 10

Commentaire: Black Ops 3 offre une expérience de tir futuriste intense, plongeant les joueurs dans un monde rempli de technologies avancées et de combats frénétiques. Les graphismes modernes et les mécaniques de

jeu fluides contribuent à l'immersion, tandis que le mode multijoueur compétitif et le mode zombies addictif ajoutent une grande variété de gameplay. Cependant, le scénario de la campagne solo peut sembler confus et la progression peut sembler un peu lente dans certains aspects. Malgré cela, Black Ops 3 reste un titre solide pour les fans de la série Call of Duty et les amateurs de jeux de tir futuristes.

Titre: Black Ops 4 : Une évolution en demi-teinte de la franchise

Note: 7 sur 10

Commentaire: Black Ops 4 propose une évolution intéressante de la franchise avec l'introduction de nouveaux modes de jeu et la suppression de la campagne solo au profit d'un mode Battle Royale, intitulé "Blackout". Les mécaniques de jeu restent solides, offrant des combats intenses et rapides, notamment dans le mode multijoueur classique. Cependant, la suppression de la campagne solo déçoit ceux qui appréciaient l'expérience scénaristique de la franchise. De plus, le mode Blackout peut souffrir de problèmes d'équilibrage et de stabilité, ce qui peut affecter l'expérience de jeu globale. Malgré ses nouvelles idées, Black Ops 4 n'a pas réussi à surpasser les attentes élevées des fans de la série et se positionne comme un épisode en demi-teinte.

fais pareil avec black ops 2

ChatGPT

Titre: Black Ops 2 : Une réussite incontournable de la franchise

Note: 9 sur 10

Commentaire: Black Ops 2 est un incontournable de la franchise Call of Duty, offrant une expérience de tir captivante et une histoire qui se déroule dans deux périodes temporelles différentes. Le scénario bien construit, combiné à des personnages mémorables, ajoute une profondeur significative à la campagne solo. Le mode multijoueur compétitif reste addictif et offre une variété de modes de jeu pour satisfaire tous les types de joueurs. De plus, le mode zombie propose des défis stimulants et une jouabilité infinie. Bien que certains pourraient critiquer le manque d'innovation en termes de mécaniques de jeu, Black Ops 2 reste un titre solide qui a su captiver les fans de la franchise grâce à son gameplay solide et à son immersion dans un univers militaire intense.



# Installation

## Composer

Pour commencer, tapez la commande suivante : « `curl -sS https://getcomposer.org/installer -o composer-setup.php` ».

Ensuite, tapez la commande suivante afin d'installer Composer de manière globale : « `sudo php composer-setup.php --install-dir=/usr/local/bin --filename=composer` »

Vous pouvez ensuite relancez votre Ubuntu et tapez « `composer --version` » pour vérifier que composer est bien installé.

## Simple-router

Ensuite il est possible de suivre la documentation qui explique de l'installation à l'utilisation de simple-router ou continuer cette documentation qui explique chaque étape de l'installation et configuration :

<https://packagist.org/packages/pecee/simple-router>

Voici la commande à taper pour installer le router dans le projet : « `composer require pecee/simple-router` ».

En cas de problème avec l'installation il est possible de récupérer les fichiers générés directement dans ce GitHub :

<https://github.com/skipperbent/simple-php-router>

## Autoload

Autoload permet de charger automatiquement les classes sans avoir à les inclure via « `require_once` » ou « `include_once` » .

Une fois simple-router installé il est possible d'intégrer autoload en suivant ces 2 étapes :

« `sudo apt-get update -y` »

« `sudo apt-get install -y php-autoload` »

Pour mettre à jour il suffit de taper cette commande :

« `composer dump-autoload` »

# Mise en place d'un Virtual Host

## Création

Pour créer un virtual host il faut aller dans le serveur et taper la commande :

« `sudo vim /etc/apache2/sites-available/[nom_du_domaine].conf` »

Dans ce fichier il faut ajouter ce code et changer les éléments entre crochet en fonction du projet.

```
<VirtualHost *: [Port]>
    DocumentRoot ["Chemin jusqu'à l'index"]
    ServerName [www.example.com]
</VirtualHost>
```

Une fois terminé il faut faire un lien symbolique vers « sites-enabled ». Pour ce faire il faut aller dans le ce dossier puis taper la commande :

« `sudo a2ensite [nom_du_domaine].conf` »

## apache2.conf

Ensuite il faut aller dans le fichier apache2.conf en tapant la commande suivante :

« `sudo vim /etc/apache2/apache2.conf` »

Dans ce fichier il faut ajouter une balise Directory comme ceci :

```
<Directory ["Même chemin que DocumentRoot"]>
    AllowOverride all
    Require all granted
</Directory>
```

Toujours dans ce fichier il faut ajouter (pas obligatoire) :

```
ServeurName [exemple.com]
```

## Hosts

Après il faut ouvrir le fichier hosts en administrateur qui se trouve dans le chemin suivant :

« C:\Windows\System32\drivers\etc »

Une fois ouvert ajouter à ce fichier :

```
127.0.0.1 [exemple.com]
```

Pour écouter le port, il faut aller dans « /etc/apache2/ports.conf » et ajouter une ligne avec votre port :

```
Listen [port]
```

Commande pour voir les ports qui sont ouverts :

« sudo apache2ctl -S »

# Configuration fichiers Autoload et Pecee/simple-router

## autoload-psr4.php

Voici le code de base :

```
return array(  
    'Pecee\\' => array($vendorDir . '/pecee/simple-router/src/Pecee'),  
);
```

Voici à quoi ressemble le code après modification :

```
return array(  
    'projet_exemple\\' => array($baseDir . '/src'),  
    'Pecee\\' => array($vendorDir . '/pecee/simple-router/src/Pecee'),  
);
```

La modification permet d'ajouter un deuxième préfixe pour créer un nouveau namespace spécifique au projet.

## autoload\_static.php

Remplacez:

```
public static $prefixLengthsPsr4 = array (  
    'p' =>  
        array (  
            'Pecee\\' => 6,  
        ),  
);  
  
public static $prefixDirsPsr4 = array (  
    'Pecee\\' =>  
        array (  
            0 => __DIR__ . '/../.' . '/pecee/simple-router/src/Pecee',  
        ),  
);
```

Par:

```
public static $prefixLengthsPsr4 = array (
    '[premiere lettre du projet]' =>
        array (
            'projet_exemple\\' => 15,
        ),
    'P' =>
        array (
            'Pecee\\' => 6,
        ),
);

public static $prefixDirsPsr4 = array (
    'projet_exemple\\' =>
        array (
            0 => __DIR__ . '/../..' . '/src',
        ),
    'Pecee\\' =>
        array (
            0 => __DIR__ . '/../' . '/pecee/simple-router/src/Pecee',
        ),
);
```

Cette modification de l'autoload\_static.php sert à ajouter un nouveau préfixe de chargement de classes pour le projet, qui permettra de charger les classes du projet à partir d'un dossier spécifique.

## .htaccess

Si la documentation d'installation n'a pas été suivie, il ne faut pas oublier de créer un fichier .htaccess dans le dossier public qui permet de rediriger vers la page index constamment. Pour ce faire, il faut ajouter ces lignes :

RewriteEngine on

RewriteCond %{SCRIPT\_FILENAME} !-f

RewriteCond %{SCRIPT\_FILENAME} !-d

RewriteCond %{SCRIPT\_FILENAME} !-l

RewriteRule ^(.\*)\$ index.php/\$1

## composer.json

Il faut ajouter :

```
"autoload": {  
    "psr-4": {  
        "projet_exemple\\": "src/"  
    }  
},
```

## index.php

Le fichier index devrait ressembler à ça :

```
<?php  
  
require '../vendor/autoload.php';  
require '../routes/routes.php';  
  
use Pecee\\SimpleRouter\\SimpleRouter as Router;  
  
// Enregistrement du namespace par défaut des controllers  
Router::setDefaultNamespace('\\projet_exemple\\Controlleurs');  
  
// Lancement du router  
Router::start();
```

## routes.php

Le fichier routes.php doit être mis dans un dossier routes par exemple à la racine du projet. Il contient tout le routage et les appellent aux contrôleurs.

Voici un exemple d'utilisation :

Dans cette ligne on utilise la classe « AccueilControlleur » pour pouvoir appeler les fonctions qui sont dedans.

```
use projet_exemple\\Controlleurs\\AccueilControlleur;
```

Ensuite il est possible de faire le routage comme dans cet exemple :

```
Router::get('/', [AccueilControlleur::class, 'accueil']);
```

Cette ligne permet de dire que quand on arrive sur le site ou que la page est « / » on appelle le contrôleur accueil puis on appelle la fonction accueil qui est dans le contrôleur.



# Code source

## index.php

```
<?php

require '../vendor/autoload.php';
require '../routes/routes.php';

use Pecee\SimpleRouter\SimpleRouter as Router;

session_start();

// Enregistrement du namespace par défaut des contrôleurs
Router::setDefaultNamespace('\ShareGames\Controllers');

// Lancement du router
Router::start();
```

## routes.php

```
<?php

use Pecee\SimpleRouter\SimpleRouter as Router;
use ShareGames\Controllers\HomeController;
use ShareGames\Controllers\RegisterController;
use ShareGames\Controllers>LoginController;
use ShareGames\Controllers\LogoutController;
use ShareGames\Controllers\ManagementGamesController;
use ShareGames\Controllers\CreateGameController;
use ShareGames\Controllers\DetailsGameController;
use ShareGames\Controllers\DeleteController;
use ShareGames\Controllers\UpdateController;
use ShareGames\Controllers\GamesController;
use ShareGames\Controllers\ProfileController;
use ShareGames\Controllers>PasswordForgotController;

Router::form('/', [HomeController::class, 'Home']);
Router::form('/inscription', [RegisterController::class, 'Register']);
Router::form('/connexion', [LoginController::class, 'Login']);
Router::get('/deconnexion', [LogoutController::class, 'Logout']);
Router::form('/gestionJeux', [ManagementGamesController::class, 'ManagementGames']);
Router::form('/creationJeu', [CreateGameController::class, 'CreateGame']);
Router::form('/detailsJeu', [DetailsGameController::class, 'DetailsGame']);
```



```
Router::get('/supprimerAvis', [DeleteController::class, 'DeleteOpinion']);
Router::get('/supprimerJeu', [DeleteController::class, 'DeleteGame']);
Router::form('/modifierAvis', [UpdateController::class, 'UpdateOpinion']);
Router::form('/modifierJeu', [UpdateController::class, 'UpdateGame']);
Router::form('/jeux', [GamesController::class, 'Games']);
Router::form('/profil', [ProfileController::class, 'Profile']);
Router::form('/modifierProfil', [UpdateController::class, 'UpdateUser']);
Router::form('/mdpOublier', [LoginController::class, 'PasswordForgot']);
```

## composer.json

```
{
    "autoload": {
        "psr-4": {
            "shareGames\\": "src/"
        }
    },
    "authors": [
        {
            "name": "Billegas Lucas",
            "email": "lucas.bllys@eduge.ch"
        }
    ],
    "require": {
        "pecee/simple-router": "5.3.0.3"
    }
}
```

## config-exemple.php

```
<?php
/**
 * Fichier de configuration pour la base de données
 *
 * Billegas Lucas
 * 27.04.2023
 */

// Initialisation des constantes
define('DB_HOST', '');
define('DB_NAME', '');
define('DB_USER', '');
define('DB_PASS', '');
```

## .htaccess

```
RewriteEngine on
RewriteCond %{SCRIPT_FILENAME} !-f
RewriteCond %{SCRIPT_FILENAME} !-d
RewriteCond %{SCRIPT_FILENAME} !-l
RewriteRule ^(.*)$ index.php/$1
```

## Models

### ConnexionDB.php

```
<?php
/**
 * Class pour la connexion à la base de données
 *
 * Billegas Lucas
 * 27.04.2023
 */

namespace ShareGames\Models;

use PDO;

require_once "../src/config.php";

class ConnexionDB
{
    /**
     * Connexion à la base de données
     */
    public static function Db()
    {
        static $pdo = null;

        if ($pdo == null) {
            $dsn = "mysql:host=" . DB_HOST . ";dbname=" . DB_NAME . ";charset=utf8";

            $pdo = new PDO($dsn, DB_USER, DB_PASS);

            $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            $pdo->setAttribute(PDO::ATTR_EMULATE_PREPARES, false);
        }

        return $pdo;
    }

    /**
     * Execute la requête
     *
     * @param string $sql Requete sql
     */
}
```

```
* @param string $param Les différents paramètres de la requete
*/
public static function DbRun($sql, $param = null)
{
    $statement = self::Db()->prepare($sql);

    $result = $statement->execute($param);

    return $statement;
}
}
```

## GamesModel.php

```
<?php

/**
 * Class model pour la gestion des fiches de jeux
 *
 * Billegas Lucas
 * 02.05.2023
 */

namespace ShareGames\Models;

use PDO;

class GamesModel
{
    const LIMIT_GET_TOP_GAME = 10;

    /**
     * Création d'une fiche de jeu
     *
     * @param date $date Le nom d'utilisateur
     * @param text $description La description de la fiche
     * @param string $title Le titre de la fiche
     * @param string $fileName Le nom de l'image
     * @param string $platform Le nom de la plateforme
     */
    public static function CreateGame($date, $description, $title, $fileName,
    $platform)
    {
        $sql = "INSERT INTO jeux (date, description, titre, vignette, plateforme)
VALUES (?, ?, ?, ?, ?)";
```

```
$param = [$date, $description, $title, $fileName, $platform];
ConnexionDB::DbRun($sql, $param);

return ConnexionDB::Db()->lastInsertId();
}

/**
 * Donne au fichier un nom unique aléatoire
 *
 * @param string $fileName Le nom du fichier
 *
 * @return string le nouveau nom
 */
public static function NameFileRandom($fileName)
{
    $extension = explode('.', $fileName)[1];

    $newName = uniqid();

    $newName .= "." . $extension;
    return $newName;
}

/**
 * Récupérer les fiches de jeu de la plus récente à la plus ancienne
 *
 * @return object Les fiches de jeux
 */
public static function GetGameMoreRecently()
{
    $sql = "SELECT * FROM jeux ORDER BY date DESC, id DESC";

    $param = [];
    return ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);
}

/**
 * Récupérer une fiche de jeu avec l'id
 *
 * @param int $idGame L'id de la fiche
 *
 * @return object La fiche de jeu
 */
public static function GetGameById($idGame)
{

```

```
$sql = "SELECT * FROM jeux WHERE id = ?";

$params = [$idGame];
return ConnexionDB::DbRun($sql, $params)->fetch(PDO::FETCH_OBJ);
}

/**
 * Supprimer un jeu
 *
 * @param int $idGame L'id de la fiche
 */
public static function DeleteGame($idGame)
{
    $sql = "DELETE FROM jeux WHERE id = ?";

    $params = [$idGame];
    return ConnexionDB::DbRun($sql, $params);
}

/**
 * Récupérer tous les jeux
 */
public static function GetAllGames()
{
    $sql = "SELECT * FROM jeux";

    $params = [];
    return ConnexionDB::DbRun($sql, $params)->fetchAll(PDO::FETCH_OBJ);
}

/**
 * Récupérer les notes d'un jeu
 *
 * @param int $idGame L'id du jeu
 * @return object Les notes du jeu
 */
public static function GetMarksOfGame($idGame)
{
    $sql = "SELECT note FROM avis INNER JOIN jeux ON jeux.id = avis.jeux_id AND
jeux.id = ?";

    $params = [$idGame];
    return ConnexionDB::DbRun($sql, $params)->fetchAll(PDO::FETCH_OBJ);
}
```

```
/**
 * Calcul de la moyenne des notes d'un jeu
 *
 * @param object $marks Les notes du jeu
 * @return double La moyenne du jeu
 */
public static function AverageMarks($marks)
{
    $average = 0;
    $nbMarks = count($marks);

    foreach ($marks as $key => $mark) {
        $average += $mark->note;
    }

    if ($nbMarks != 0) {
        $average = (double)($average / $nbMarks);
    } else {
        $average = 0;
    }

    return $average;
}

/**
 * Récupérer les 10 meilleurs jeux
 *
 * @return array $topGames Le tableau des 10 meilleurs jeux
 */
public static function GetTopGames()
{
    $games = self::GetAllGames();

    $averageMarks = array();
    foreach ($games as $game) {
        $marks = self::GetMarksOfGame($game->id);

        // Vérifie si le jeu est noté
        if (count($marks) > 0) {
            $average = self::AverageMarks($marks);
            $averageMarks[$game->id] = $average;
        }
    }

    // Ordonne le tableau dans l'ordre décroissant
    arsort($averageMarks);
}
```

```
$topGames = array();
$i = 0;
foreach ($averageMarks as $gameId => $averageMark) {
    // Attribue le jeu dans un tableau
    $topGames[] = self::GetGameById($gameId);
    $i++;
    // Arrête d'ajouter au 10ème
    if ($i == self::LIMIT_GET_TOP_GAME) {
        break;
    }
}

return $topGames;
}

/**
 * Modifier le jeu
 *
 * @param date $date la date
 * @param string $title le titre
 * @param text $description la description
 * @param string $fileName le nom de la vignette
 * @param string $platform la plateforme
 * @param int $idGame L'id du jeu
 * @return void
 */
public static function UpdateGame($date, $title, $description, $fileName,
$platform, $idGame)
{
    $sql = "UPDATE jeux SET date = ?, description = ?, titre = ?, vignette = ?,
plateforme = ? WHERE id = ?";

    $param = [$date, $description, $title, $fileName, $platform, $idGame];
    return ConnexionDB::DbRun($sql, $param);
}

/**
 * Récupérer les jeux en fonction d'une recherche sur le titre ou la description
 *
 * @param string $search la recherche
 * @return object Les jeux correspondant à la recherche
 */
public static function SearchGamesByTitleOrDescription($search)
{
    $sql = "SELECT * FROM jeux WHERE description LIKE ? OR titre LIKE ? ";
```



```
$param = ["%$search%", "%$search%"];
return ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);
}

/**
 * Récupérer les jeux que l'utilisateur a commenté
 *
 * @param [type] $idUser L'id de l'utilisateur
 * @return object Les jeux commentés par l'utilisateur
 */
public static function GetGamesOfUserComment($idUser)
{
    $sql = "SELECT DISTINCT jeux.id FROM jeux INNER JOIN avis ON jeux.id =
avis.jeux_id INNER JOIN utilisateurs ON utilisateurs.id = avis.utilisateurs_id AND
utilisateurs.id = ?";

    $param = [$idUser];
    return ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);
}

/**
 * Tri les jeux par la moyenne des notes
 *
 * @return array $sortedGames Le tableau des jeux triés
 */
public static function SortedGamesByAverageMark()
{
    $games = self::GetAllGames();

    $averageMarks = array();
    foreach ($games as $game) {
        $marks = self::GetMarksOfGame($game->id);
        if (count($marks) > 0) {
            $average = self::AverageMarks($marks);
            $averageMarks[$game->id] = $average;
        } else {
            $averageMarks[$game->id] = 0;
        }
    }

    // Tri décroissant des moyennes
    arsort($averageMarks);

    $sortedGames = array();
    foreach ($averageMarks as $gameId => $averageMark) {
```

```
        $sortedGames[] = self::GetGameById($gameId);
    }

    return $sortedGames;
}

/**
 * Récupérer un certain nombre de jeux
 *
 * @param int $offset L'index du premier jeu a récupérer
 * @param int $next Le nombre de jeux à récupérer
 * @param string $idGames Les identifiants des jeux
 * @return void
 */
public static function GetGamesByLimit($offset, $next, $idGames){
    $sql = "SELECT * FROM jeux WHERE id IN ($idGames) ORDER BY FIELD(id, $idGames)
LIMIT ?, ?";

    $param = [$offset, $next];
    return ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);
}
}
```

### OpinionsModel.php

```
<?php

/**
 * Class model pour la gestion des avis
 *
 * Billegas Lucas
 * 03.05.2023
 */

namespace ShareGames\Models;

use PDO;

class OpinionsModel
{
    /**
     * Créer un nouvel avis
     *
     * @param string $title Le titre du commentaire
     * @param string $description La description du commentaire
     * @param int $mark La note du commentaire
     * @param int $idUser L'id de l'utilisateur
     */
}
```

```
* @param int $idGame L'id du jeu
* @param date $title Le titre du commentaire
*/
public static function CreateOpinion($title, $description, $mark, $idUser, $idGame,
$date)
{
    $sql = "INSERT INTO avis (date, titre, description, note, utilisateurs_id,
jeux_id) VALUES (?, ?, ?, ?, ?, ?)";

    $param = [$date, $title, $description, $mark, $idUser, $idGame];
    return ConnexionDB::DbRun($sql, $param);
}

/**
 * Récupère tous les avis d'un jeu dans l'ordre chronologique
 *
 * @param int $idGame L'id du jeu
 *
 * @return object Les avis
 */
public static function GetAllOpinions($idGame){
    $sql = "SELECT * FROM avis WHERE jeux_id = ? ORDER BY date DESC, id DESC";

    $param = [$idGame];
    return ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);
}

/**
 * Supprimer un avis
 *
 * @param int $idOpinion L'id de l'avis
 */
public static function DeleteOpinion($idOpinion){
    $sql = "DELETE FROM avis WHERE id = ?";

    $param = [$idOpinion];
    return ConnexionDB::DbRun($sql, $param);
}

/**
 * Modifier un avis
 *
 * @param int $idOpinion L'id de l'avis
 * @param string $title Le titre de l'avis
 * @param text $description La description de l'avis
 * @param int $mark La note de l'avis
```

```
* @param date $date La date de l'avis
* @return void
*/
public static function UpdateOpinion($idOpinion, $title, $description, $mark,
$date){
    $sql = "UPDATE avis SET date = ?, titre = ?, description = ?, note = ? WHERE id
= ?";

    $param = [$date, $title, $description, $mark, $idOpinion];
    return ConnexionDB::DbRun($sql, $param);
}

/**
 * Compte les commentaires de l'utilisateur
 *
 * @param int $idUser l'id de l'utilisateur
 * @return int Le nombre de commentaires
 */
public static function CountCommentaryOfUser($idUser){
    $sql = "SELECT * FROM avis WHERE utilisateurs_id = ?";

    $param = [$idUser];
    return count(ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ));
}

/**
 * Récupérer un avis par rapport à son identifiant
 *
 * @param int $idOpinion L'id de l'avis
 * @return object L'avis
 */
public static function GetOpinionById($idOpinion){
    $sql = "SELECT * FROM avis WHERE id = ?";

    $param = [$idOpinion];
    return ConnexionDB::DbRun($sql, $param)->fetch(PDO::FETCH_OBJ);
}
}
```

TypesModel.php

&lt;?php

```
/**
 * Class model pour la gestion des types
 *
 * Billegas Lucas
 * 03.05.2023
 */

namespace ShareGames\Models;

use PDO;

class TypesModel
{
    /**
     * Récupérer tous les types de jeux
     */
    public static function GetAllTypes()
    {
        $sql = "SELECT * FROM types";

        $param = [];
        return ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);
    }

    /**
     * Récupérer l'identifiant du type par rapport a son nom
     *
     * @param string $type Le nom du type
     * @return object l'id du type
     */
    public static function GetIdTypeByName($type){
        $sql = "SELECT id FROM types WHERE type = ?";

        $param = [$type];
        return ConnexionDB::DbRun($sql, $param)->fetch(PDO::FETCH_OBJ);
    }

    /**
     * Donne un type à un jeu
     *
     * @param [type] $idGame L'id du jeu
     * @param [type] $idType L'id du type de jeu
     * @return void
     */
}
```

```
public static function GiveTypeToGame($idGame, $idType){
    $sql = "INSERT INTO jeux_has_types(jeux_id, types_id) VALUES (?,?)";

    $param = [$idGame, $idType];
    return ConnexionDB::DbRun($sql, $param);
}

/**
 * Réuperer les types d'un jeu
 *
 * @param [type] $idGame L'id du jeu
 * @return array $types Les types du jeu
 */
public static function GetTypesOfGame($idGame){
    $sql = "SELECT types_id FROM jeux_has_types WHERE jeux_id = ?";

    $param = [$idGame];
    $types_id = ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);

    $types = array();

    foreach ($types_id as $key => $type_id) {
        $sql = "SELECT type FROM types WHERE id = ?";

        $param = [$type_id->types_id];

        array_push($types, ConnexionDB::DbRun($sql, $param)-
>fetch(PDO::FETCH_OBJ));
    }

    return $types;
}

/**
 * Supprimer le type d'un jeu
 *
 * @param [type] $idGame L'id du jeu
 * @param [type] $idType L'id du type
 * @return void
 */
public static function DeleteTypeOfGame($idGame, $idType){
    $sql = "DELETE FROM jeux_has_types WHERE jeux_id = ? AND types_id = ?";

    $param = [$idGame, $idType];
    return ConnexionDB::DbRun($sql, $param);
}
```

```
}  
}
```

## UsersModel.php

```
<?php  
  
/**  
 * Class model pour la gestion d'utilisateurs  
 *  
 * Billegas Lucas  
 * 02.05.2023  
 */  
  
namespace ShareGames\Models;  
  
use PDO;  
  
class UsersModel  
{  
    const USER_AVATAR_DEFAULT = "User_Avatar_Default.png";  
    const CHAR_PWD_DEFAULT = 8;  
  
    /**  
     * Création d'un utilisateur  
     *  
     * @param string $name Le nom d'utilisateur  
     * @param string $password Le mot de passe hasher  
     * @param string $fileName Le nom du fichier  
     * @param int $admin Utilisateur admin ou pas  
     */  
    public static function CreateUser($name, $password, $fileName, $admin)  
    {  
        $sql = "INSERT INTO utilisateurs (login, password, avatar, est_admin) VALUES  
(?,?,?,?)";  
  
        $param = [$name, $password, $fileName, $admin];  
        return ConnexionDB::DbRun($sql, $param);  
    }  
  
    /**  
     * Verifie si le nom d'utilisateur est déjà pris  
     *  
     * @param string $name Le nom d'utilisateur  
     * @return bool Si l'utilisateur existe ou pas  
     */  
}
```

```
*/
public static function VerifyIfUserExist($name)
{
    $sql = "SELECT * FROM utilisateurs WHERE login = ?";

    $param = [$name];
    $result = ConnexionDB::DbRun($sql, $param)->fetchAll(PDO::FETCH_OBJ);

    if (count($result) > 0) {
        return true;
    } else {
        return false;
    }
}

/**
 * Renomme l'avatar pour le rendre unique avec le nom d'utilisateur
 *
 * @param string $fileName Le nom du fichier
 * @param string $nameUser Le nom d'utilisateur
 * @return string $fileName Le nouveau nom du fichier
 */
public static function RenameAvatar($fileName, $nameUser)
{
    $extension = explode(".", $fileName)[1];
    $fileName = $nameUser . "." . $extension;

    return $fileName;
}

/**
 * Récupère l'utilisateur depuis le nom
 *
 * @param string $name Le nom d'utilisateur
 * @return object L'utilisateur
 */
public static function GetUserByName($name)
{
    $sql = "SELECT * FROM utilisateurs WHERE login = ?";

    $param = [$name];
    return ConnexionDB::DbRun($sql, $param)->fetch(PDO::FETCH_OBJ);
}

/**
 * Récupère l'utilisateur depuis l'identifiant
```



```
*
* @param string $idUser l'id de l'utilisateur
* @return object L'utilisateur
*/
public static function GetUserById($idUser)
{
    $sql = "SELECT * FROM utilisateurs WHERE id = ?";

    $param = [$idUser];
    return ConnexionDB::DbRun($sql, $param)->fetch(PDO::FETCH_OBJ);
}

/**
 * Change le mot de passe de l'utilisateur
 *
 * @param string $newPassword Le nouveau mot de passe
 * @param int $idUser L'id de l'utilisateur
 */
public static function ChangePassword($newPassword, $idUser)
{
    $sql = "UPDATE utilisateurs SET password = ? WHERE id = ?";
    $param = [$newPassword, $idUser];

    return ConnexionDB::DbRun($sql, $param);
}

/**
 * Change l'avatar de l'utilisateur
 *
 * @param [type] $newAvatar Le nouveau avatar
 * @param [type] $idUser L'id de l'utilisateur
 * @return void
 */
public static function ChangeAvatar($newAvatar, $idUser)
{
    $sql = "UPDATE utilisateurs SET avatar = ? WHERE id = ?";
    $param = [$newAvatar, $idUser];

    return ConnexionDB::DbRun($sql, $param);
}

/**
 * Génère un mot de passe aléatoire
 *
 * @param int $length La taille du mot de passe
 * @return string $password Le mot de passe
 */
```

```
*/  
public static function GeneratePassword($length = self::CHAR_PWD_DEFAULT)  
{  
  
    $characters = array_merge(range('a', 'z'), range('A', 'Z'), range(0, 9));  
    $password = "";  
  
    for ($i = 0; $i < $length; $i++) {  
        $password .= $characters[rand(0, count($characters) - 1)];  
    }  
    return $password;  
}  
}
```

## Controllers

### HomeController.php

```
<?php  
  
/**  
 * Contrôleur qui permet la gestion de l'accueil  
 *  
 * Billegas Lucas  
 * 27.04.2023  
 */  
  
namespace ShareGames\Controllers;  
  
use ShareGames\Models\GamesModel;  
  
class HomeController  
{  
    const LIMIT_GAMES_RECENTLY = 5;  
    const HALF_TOP_GAMES = 6;  
  
    /**  
     * Accueil  
     *  
     * @return void  
     */  
    public function Home()  
    {  
        $topGames = GamesModel::GetTopGames();  
        $cardsTopGames = self::CardTopGames($topGames);  
        $cardsMoreRecently = self::CardGamesMoreRecently();  
    }  
}
```

```
        require_once "../src/Views/homeView.php";
    }

    /**
     * Carte pour les 10 meilleurs jeux
     *
     * @param [type] $topGames Les 10 meilleurs jeux
     * @return string $output L'affichage de la carte
     */
    public function CardTopGames($topGames)
    {
        $nbCard = 1;
        $output = "<div class='row align-items-stretch retro-layout-alt'><div
class='two-col d-block d-md-flex justify-content-between' style='margin-bottom: 1%;'>";
        foreach ($topGames as $game) {
            // Affichage 5 cartes par lignes
            if ($nbCard == self::HALF_TOP_GAMES) {
                $output .= "</div><div class='two-col d-block d-md-flex justify-
content-between'>";
            }
            $output .= self::CardNormaly($game);
            $nbCard++;
        }

        $output .= "</div></div>";
        return $output;
    }

    /**
     * Affichage des jeux les plus récents
     *
     * @return string $output L'affichage des jeux les plus récents
     */
    public function CardGamesMoreRecently()
    {
        $output = "<div class='row align-items-stretch retro-layout-alt'>";
        $nbCard = 0;
        $games = GamesModel::GetGameMoreRecently();

        foreach ($games as $game) {
            $nbCard++;
            if ($nbCard > self::LIMIT_GAMES_RECENTLY) {
                break;
            }
            if ($nbCard == 1) {
```

```

        $output .= self::FirstCard($game);
        $output .= "<div class='col-md-7'>";
    } else if ($nbCard % 2 == 0) {
        $output .= "<div class='two-col d-block d-md-flex justify-content-
between' style='margin-bottom: 1%;'>";
    }

    $output .= self::CardNormaly($game);

    if ($nbCard == 3 || $nbCard == 5) {
        $output .= "</div>";
    }
}

$output .= "</div></div>";
return $output;
}

/**
 * Affichage du jeu le plus récent
 *
 * @param object $game Le jeu
 * @return string $output L'affichage du jeu le plus récent
 */
public function FirstCard($game)
{
    // Change le format de la date
    $dateFormat = date("d-m-Y", strtotime($game->date));
    return "<div class='col-md-5 order-md-0'>
        <a href='detailsJeu?idGame=" . $game->id . "' class='hentry img-1
h-100 gradient'>
            <div class='featured-img' style='background-image:
url(assets/images/" . $game->vignette . ");'></div>
            <div class='text'>
                <span>" . $dateFormat . "</span>
                <h2>" . $game->titre . "</h2>
            </div>
        </a>
    </div>";
}

/**
 * Affichage du 2 au 5ème jeux les plus récents
 *
 * @param object $game Le jeu
 * @return string $output L'affichage du jeu
 */

```

```
public function CardNormaly($game)
{
    // Change le format de la date
    $dateFormat = date("d-m-Y", strtotime($game->date));

    return "<a href='detailsJeu?idGame=" . $game->id . "' class='hentry v-height
img-2 ms-auto float-end gradient'>
    <div class='featured-img' style='background-image: url(assets/images/" . $game-
>vignette . "');></div>
    <div class='text text-sm'>
        <span>" . $dateFormat . "</span>
        <h2>" . $game->titre . "</h2>
    </div>
</a>";
}
```

## LoginController.php

```
<?php

/**
 * Contrôleur qui permet la gestion de la connexion
 *
 * Billegas Lucas
 * 27.04.2023
 */

namespace ShareGames\Controllers;

use ShareGames\Models\UsersModel;

class LoginController
{
    /**
     * Connexion
     *
     * @return void
     */
    public function Login()
    {
        define("ADMIN_DEFAULT", 0);

        $message = "";

        if (filter_input(INPUT_POST, 'login')) {
            $name = filter_input(INPUT_POST, 'name');
```

```
$password = filter_input(INPUT_POST, 'password');

$user = UsersModel::getUserByName($name);
if ($user != null) {
    // Vérifie que les mots de passe correspondent
    if (password_verify($password, $user->password)) {

        $_SESSION['name'] = $user->login;
        $_SESSION['idUser'] = $user->id;
        $_SESSION['userIsConnected'] = true;

        if ($user->est_admin != ADMIN_DEFAULT) {
            $_SESSION['admin'] = true;
        } else {
            $_SESSION['admin'] = false;
        }

        header("Location: /");
        exit;
    } else {
        $message = "Impossible de se connecter au compte.";
    }
} else {
    $message = "Impossible de se connecter au compte.";
}

require_once "../src/Views/loginView.php";
}

/**
 * Oublie du mot de passe
 *
 * @return string Le message de retour
 */
public function PasswordForgot()
{
    header('Content-Type: application/json');

    $name = filter_input(INPUT_POST, "name");
    if ($name != "") {
        if (UsersModel::getUserByName($name) != null) {
            $newPwd = UsersModel::GeneratePassword();

            UsersModel::ChangePassword(password_hash($newPwd, PASSWORD_DEFAULT),
UsersModel::getUserByName($name)->id);
```

```
        echo json_encode("Votre nouveau mot de passe est : " . $newPwd);
        exit;
    }
    else{
        echo json_encode("Aucun utilisateur avec ce nom");
        exit;
    }

}
else{
    echo json_encode("Veuillez remplir le champs nom");
    exit;
}
}
}
```

### RegisterController.php

```
<?php
/**
 * Contrôleur qui permet la gestion de l'inscription
 *
 * Billegas Lucas
 * 27.04.2023
 */

namespace ShareGames\Controllers;

use ShareGames\Models\UsersModel;

class RegisterController{
    /**
     * Inscription
     *
     * @return void
     */
    public function Register(){

        define("ADMIN_DEFAULT", 0);

        $message = "";

        if (filter_input(INPUT_POST, 'register')) {
            $name = filter_input(INPUT_POST, 'name');
            $password = filter_input(INPUT_POST, 'password');
            $passwordConfirm = filter_input(INPUT_POST, 'confirmPassword');
            $fileName = $_FILES['avatar']['name'];
```

```
// Vérifie que les mots de passe correspondent
if ($password === $passwordConfirm) {
    // Hash du mot de passe
    $hashedPassword = password_hash($password, PASSWORD_DEFAULT);

    if (!UsersModel::VerifyIfUserExist($name)) {

        if ($fileName != null) {
            $fileName = UsersModel::RenameAvatar($fileName, $name);
            $file_tmp = $_FILES['avatar']['tmp_name'];
            move_uploaded_file($file_tmp, "assets/images/" . $fileName);
        }

        UsersModel::CreateUser($name, $hashedPassword, $fileName,
ADMIN_DEFAULT);

        header("Location: /connexion");
        exit;
    }
    else{
        $message = "Le nom d'utilisateur est déjà utilisé.";
    }
}
else{
    $message = "Les mots de passes ne sont pas identiques.";
}

require_once "../src/Views/registerView.php";
}
}
```



## ProfileController.php

```
<?php

/**
 * Contrôleur qui permet la gestion du profil
 *
 * Billegas Lucas
 * 08.05.2023
 */

namespace ShareGames\Controllers;

use ShareGames\Models\GamesModel;
use ShareGames\Models\OpinionsModel;
use ShareGames\Models\UsersModel;

class ProfileController
{
    public function Profile()
    {
        if (isset($_SESSION['idUser'])) {

            $displayGames = "";

            $user = UsersModel::getUserById($_SESSION['idUser']);

            if ($user->avatar == null) {
                $avatar = UsersModel::USER_AVATAR_DEFAULT;
            }
            else{
                $avatar = $user->avatar;
            }

            $idGames = GamesModel::getGamesOfUserComment($_SESSION['idUser']);
            $nbGames = count($idGames);

            $nbCommentary = OpinionsModel::countCommentaryOfUser($_SESSION['idUser']);

            if ($nbGames > 0) {
                $displayGames = self::displayGames($idGames);
            }

            require_once "../src/Views/profileView.php";
        } else {
```

```
        header("Location: /");
        exit;
    }
}
/**
 * Affichage des jeux
 *
 * @param array $idGames Les identifiants des jeux à afficher
 * @return string $output L'affichage des jeux
 */
public function DisplayGames($idGames)
{
    $countGames = 0;
    $output = "";
    foreach ($idGames as $idGame) {

        $game = GamesModel::GetGameById($idGame->id);

        if ($countGames % 2 == 0) {
            $output .= "<div class='row g-2'>";
        }

        $output .= "<div class='col mb-2'><a href='detailsJeu?idGame=$idGame-";
    }
    $output .= ">id'>";
    if ($game->vignette == null) {
        $output .= "<img src='assets/images/'. UsersModel::USER_AVATAR_DEFAULT";
    }
    else{
        $output .= "<img src='assets/images/$game->vignette'";
    }

    $output .= "alt='image 1' class='w-100 h-100 rounded-3'></a>";
    $output .= "</div>";

    if ($countGames % 2 != 0) {
        $output .= "</div>";
    }

    $countGames++;
}

return $output;
}
```

```
}
```

## DetailsGameController.php

```
<?php

/**
 * Contrôleur qui permet la gestion des détails d'un jeu
 *
 * Billegas Lucas
 * 02.05.2023
 */

namespace ShareGames\Controllers;

use ShareGames\Models\GamesModel;
use ShareGames\Models\OpinionsModel;
use ShareGames\Models\TypesModel;
use ShareGames\Models\UsersModel;

class DetailsGameController
{
    /**
     * Détails du jeu
     *
     * @return void
     */
    public function DetailsGame()
    {
        $idGame = filter_input(INPUT_GET, "idGame");

        $game = GamesModel::GetGameById($idGame);
        $opinions = OpinionsModel::GetAllOpinions($idGame);

        $nbOpinions = count($opinions);

        $displayOpinions = "";

        if ($nbOpinions > 0) {
            $displayOpinions = self::DisplayOpinions($opinions, $idGame);
        }

        $listTypes = self::DisplayTypes($idGame);

        // Poster un avis
        if (filter_input(INPUT_POST, "btnSubmitOpinion")) {
```

```
$title = filter_input(INPUT_POST, "title");
$mark = filter_input(INPUT_POST, "mark", FILTER_VALIDATE_INT);
$description = filter_input(INPUT_POST, "description");
$date = date("Y-m-d");

OpinionsModel::CreateOpinion($title, $description, $mark,
$_SESSION['idUser'], $idGame, $date);
header("Location: detailsJeu?idGame=" . $idGame);
exit;
}

require "../src/Views/detailsGameView.php";
}

/**
 * Affichage des avis
 *
 * @param object $opinions les avis à afficher
 * @param int $idGame L'id du jeu
 * @return string $output L'affichage des avis
 */
public function DisplayOpinions($opinions, $idGame)
{
    $output = "<ul class='comment-list'>";
    foreach ($opinions as $opinion) {
        $user = UsersModel::GetUserById($opinion->utilisateurs_id);

        $output .= "<li class='comment'>
        <div class='vcard'>";

        if ($user->avatar == null) {
            $output .= "<img src='assets/images/" . UsersModel::USER_AVATAR_DEFAULT
            . "' alt='Image placeholder'>";
        } else {
            $output .= "<img src='assets/images/$user->avatar' alt='Image
placeholder'>";
        }

        $output .= "</div>
        <div class='comment-body'>
        <h3>$user->login</h3>
        <div class='meta'>" . date("d-m-Y", strtotime($opinion->date)) .
        "</div><span style='font-weight: bold;'>Note : $opinion->note</span><br>
        <br>
        <h5>$opinion->titre</h5>
        <p>$opinion->description</p>";
    }
}
```

```
        if (isset($_SESSION['idUser']) && $_SESSION['idUser'] == $user->id ||
isset($_SESSION['admin']) && $_SESSION['admin']) {
            $output .= "<p><a href='supprimerAvis?idOpinion=" . $opinion->id .
"&idGame=" . $idGame . "' class='reply rounded'>Supprimer</a>";
        }

        if (isset($_SESSION['idUser']) && $_SESSION['idUser'] == $user->id) {
            $output .= "<a href='modifierAvis?idOpinion=" . $opinion->id .
"&idGame=" . $idGame . "' class='reply rounded'>Modifier</a></p>";
        }
        else{
            $output .= "</p>";
        }

        $output .= "</div>
</li>";
    }
    $output .= "</ul>";
    return $output;
}

/**
 * Affichage des types du jeu dans une liste
 *
 * @param int $idGame L'id du jeu
 * @return string $output L'affichage des types
 */
public function DisplayTypes($idGame)
{
    $types = TypesModel::GetTypesOfGame($idGame);
    $output = "";

    foreach ($types as $key => $type) {

        $output .= "<li><a href='#'>$type->type</a></li>";
    }
    return $output;
}
}
```

ManagementGamesController.php

&lt;?php

```
namespace ShareGames\Controllers;

use ShareGames\Models\GamesModel;

class ManagementGamesController
{
    const LIMIT_GAME_TO_PAGE = 9;

    /**
     * Gestion des jeux
     *
     * @return void
     */
    public function ManagementGames()
    {
        $page = filter_input(INPUT_GET, "page");

        $offset = FunctionsController::GetOffset($page, self::LIMIT_GAME_TO_PAGE);

        $allGames = GamesModel::GetAllGames();

        $conditionIn = FunctionsController::GetStrIdsGames($allGames);

        $games = GamesModel::GetGamesByLimit($offset, self::LIMIT_GAME_TO_PAGE,
        $conditionIn);

        $displayGames = FunctionsController::DisplayGames($games, true);
        $displayPagination = FunctionsController::DisplayPagination($allGames,
        self::LIMIT_GAME_TO_PAGE, "gestionJeux");

        require "../src/Views/managementGamesView.php";
    }
}
```

## CreateGameController.php

```
<?php

/**
 * Contrôleur qui permet de créer une fiche de jeu
 *
 * Billegas Lucas
 * 02.05.2023
 */

namespace ShareGames\Controllers;

use ShareGames\Controllers\CreateGameController as ControllersCreateGameController;
use ShareGames\Models\GamesModel;
use ShareGames\Models\TypesModel;

class CreateGameController
{
    /**
     * Créer un jeu
     *
     * @return void
     */
    public function CreateGame()
    {
        if (!$_SESSION['admin']) {
            header("Location: /");
        } else {

            $message = "";
            $checkbox = FunctionsController::GenerateCheckbox();

            if (filter_input(INPUT_POST, 'btnCreate')) {
                $files = $_FILES['file'];
                if ($_FILES['file']['name'] != "") {
                    if (isset($_POST['types'])) {

                        $title = filter_input(INPUT_POST, 'title');
                        $description = filter_input(INPUT_POST, 'description');
                        $platform = filter_input(INPUT_POST, "platform");
                        $types = $_POST['types'];

                        $date = date("Y-m-d");

                        // Traitement de l'image
                        $fileName = $_FILES['file']['name'];
```

```
        $file_tmp = $_FILES['file']['tmp_name'];
        $fileName = GamesModel::NameFileRandom($fileName);

        move_uploaded_file($file_tmp, "assets/images/" . $fileName);

        $idGame = GamesModel::CreateGame($date, $description, $title,
$fileName, $platform);

        // Attribue les types au jeu
        foreach ($types as $key => $type) {
            $idType = TypesModel::GetIdTypeByName($type);
            TypesModel::GiveTypeToGame($idGame, $idType->id);
        }

        header("Location: gestionJeux");
        exit;
    } else {
        $message = "Veuillez sélectionner au moins un type.";
    }
    } else {
        $message = "Veuillez ajouter une image";
    }
}
}
require "../src/Views/createGameView.php";
}
}
```

### FunctionsController.php

```
<?php
/**
 * Contrôleur qui permet la gestion de différentes fonctions utilisé plusieurs fois
 dans le projet
 *
 * Billegas Lucas
 * 09.05.2023
 */

namespace ShareGames\Controllers;

use ShareGames\Models\TypesModel;
use ShareGames\Models\GamesModel;
```



```
class FunctionsController
{
    const MAX_MARK = 10;

    // Constante qui permet de savoir si c'est l'affichage de la page gestion ou la
    // page jeu
    const PAGE_DISPLAY_MANAGEMENT = true;

    /**
     * Génération de checkbox
     *
     * @return string $checkbox Les checkboxes
     */
    public static function GenerateCheckbox(){

        $types = TypesModel::GetAllTypes();
        $checkbox = "";
        foreach ($types as $type) {
            $checkbox .= "<input style='margin-left: 5px;' type='checkbox'
name='types[]' value='$type->type'>$type->type";
        }
        return $checkbox;
    }

    /**
     * Récupérer à partir de quelle ligne il faut récupérer les jeux
     *
     * @param int $page Le numéro de la page
     * @param int $limitGameToPage La limite de jeux par page
     *
     * @return int $offset Le numéro de la ligne
     */
    public static function GetOffset($page, $limitGameToPage){
        if (isset($page) && $page != 1) {
            $offset = $limitGameToPage * ($page - 1);
        }
        else{
            $page = 1;
            $offset = $limitGameToPage * ($page - 1);
        }
        return $offset;
    }

    /**
     * Affiche la pagination en fonction du nombre de jeux
     */
}
```

```
*
* @param int $page Le numéro de la page
* @param int $limitGameToPage La limite de jeux par page
* @param string $pageName Le nom de la page
*
* @return string $output L'affichage de la pagination
*/
public static function DisplayPagination($games, $limitGameToPage, $pageName){
    $output = " <div class='row text-start pt-5 border-top paginationCenter'>
        <div class='col-md-12'>
            <div class='custom-pagination'>
                <a href='$pageName?page=1'>1</a>";
    $page = 1;
    for ($i=1; $i <= count($games); $i++) {
        if ($i == $limitGameToPage * $page) {
            $page++;
            $output .= "<a href='$pageName?page=$page'>$page</a>";
        }
    }
    $output .= "</div></div></div>";
    return $output;
}

/**
* Récupérer en string la liste des ids de jeux a affiché
*
* @param object $games Les jeux
* @return string La liste des ids
*/
public static function GetStrIdsGames($games){
    $idGames = array();
    foreach ($games as $game) {
        array_push($idGames, $game->id);
    }

    return implode(',', $idGames);
}

/**
* Affichage des jeux sur la page de jeux et de la page de gestion
*
* @param object $games Les jeux à afficher
* @param bool $pageManagement Page management = true / Page jeux = false;
*
* @return string $ouptut affichage
```

```
*/
public static function DisplayGames($games, $pageManagement){
    $output = "";

    foreach ($games as $game) {

        // Récupère la moyenne des notes arrondi au 10ème (1 chiffre après la
virgule)
        $average = round(GamesModel::AverageMarks(GamesModel::GetMarksOfGame($game-
>id)), 1);

        // Récupère la moyenne des notes du jeu arrondi à l'entier
        $averageRound = round($average);

        $types = TypesModel::GetTypesOfGame($game->id);

        $dateFormat = date("d-m-Y", strtotime($game->date));

        $output .= "<div class='col-lg-4 mb-4'>
        <div class='post-entry-alt'>
            <a href='detailsJeu?idGame=$game->id' class='img-link'><img
src='assets/images/$game->vignette' alt='Image' class='img-fluid'></a>
            <div class='excerpt'>
                <h2><a href='detailsJeu?idGame=$game->id'>$game->titre</a></h2>
                <div class='post-meta align-items-center text-left clearfix'>
                    <span>$dateFormat</span>";

        foreach ($types as $type) {
            $output .= "<span style='float:right'>&nbsp; $type->type</span>";
        }

        $output .= "<br><div style='text-align: center;'>";

        if ($average != 0) {
            // Affichage des moyennes sous forme d'étoiles
            for ($i = 0; $i < self::MAX_MARK; $i++) {
                if ($i < $averageRound) {
                    $output .= "<span class='fa fa-star starChecked'></span>";
                } else {
                    $output .= "<span class='fa fa-star'></span>";
                }
            }
            $output .= "<span>&nbsp;$average/" . self::MAX_MARK . "</span>";
        } else {
            $output .= "<span>Non noté</span>";
        }
    }
}
```

```
        $output .= "</div></div><p>$game->description</p>";
        if ($pageManagement == self::PAGE_DISPLAY_MANAGEMENT) {
            $output .= "<div style='display:block;'><p><a
href='modifierJeu?idGame=$game->id' class='btn btn-sm btn-outline-primary
cardGame'>Modifier</a>";
            $output .= "<a href='supprimerJeu?idGame=$game->id' class='btn btn-
outline-danger cardGame'>Supprimer</a></p></div>";
        }
        else{
            $output .= "<p><a href='detailsJeu?idGame=$game->id' class='btn btn-sm
btn-outline-primary cardGame'>Voir plus</a></p>";
        }
        $output .= "</div></div></div>";
    }

    return $output;
}
}
```

### UpdateController.php

```
<?php

/**
 * Contrôleur qui gère les modifications
 *
 * Billegas Lucas
 * 04.05.2023
 */

namespace ShareGames\Controllers;

use ShareGames\Models\GamesModel;
use ShareGames\Models\OpinionsModel;
use ShareGames\Models\TypesModel;
use ShareGames\Models\UsersModel;

class UpdateController
{
    /**
     * Modifier l'avis
     *
     * @return void
     */
    public function UpdateOpinion()
    {
    }
```

```
$idOpinion = filter_input(INPUT_GET, "idOpinion");
$idGame = filter_input(INPUT_GET, "idGame");

$opinion = OpinionsModel::GetOpinionById($idOpinion);

$title = $opinion->titre;
$mark = $opinion->note;

$description = "";
if ($opinion->description != null) {
    $description = $opinion->description;
}

if (filter_input(INPUT_POST, "btnSubmitOpinion")) {
    $title = filter_input(INPUT_POST, "title");
    $mark = filter_input(INPUT_POST, "mark", FILTER_VALIDATE_INT);
    $description = filter_input(INPUT_POST, "description");
    $date = date("Y-m-d");

    OpinionsModel::UpdateOpinion($idOpinion, $title, $description, $mark,
$date);

    header("Location: detailsJeu?idGame=" . $idGame);
    exit;
}

require "../src/Views/updateOpinionView.php";
}

/**
 * Modifier le jeu
 *
 * @return void
 */
public function UpdateGame()
{
    if ($_SESSION['admin']) {
        $idGame = filter_input(INPUT_GET, "idGame");
        $game = GamesModel::GetGameById($idGame);

        $title = $game->titre;
        $fileName = $game->vignette;

        $description = "";
        if ($game->description != null) {
            $description = $game->description;
        }
    }
}
```

```
}

$message = "";
$checkbox = FunctionsController::GenerateCheckbox();

if (filter_input(INPUT_POST, 'btnUpdateGame')) {
    $file = $_FILES['file'];

    if (isset($_POST['types'])) {
        if ($file['name'] != "") {
            // Suppression de l'ancienne image dans le dossier
            unlink("assets/images/" . $fileName);

            $fileName = $file['name'];
            $file_tmp = $file['tmp_name'];

            $fileName = GamesModel::NameFileRandom($fileName);
            move_uploaded_file($file_tmp, "assets/images/" . $fileName);
        }

        // Récupération des champs du formulaire
        $title = filter_input(INPUT_POST, 'title');
        $description = filter_input(INPUT_POST, 'description');
        $platform = filter_input(INPUT_POST, "platform");
        $newTypes = $_POST['types'];

        $date = date("Y-m-d");

        $typesOfGame = TypesModel::GetTypesOfGame($idGame);
        $typeAlreadyExist = false;

        // Attribution des types qui ne sont pas encore dans la base de
données
        foreach ($newTypes as $key => $newType) {
            foreach ($typesOfGame as $key => $typeExist) {
                if ($typeExist->type == $newType) {
                    $typeAlreadyExist = true;
                }
            }
            if (!$typeAlreadyExist) {
                $idType = TypesModel::GetIdTypeByName($newType);
                TypesModel::GiveTypeToGame($idGame, $idType->id);
            }
            $typeAlreadyExist = false;
        }
    }
}
```

```
// Suppression des types qui n'ont pas été cochés
$typesToDelete = array();
foreach ($typesOfGame as $key => $typeExist) {
    if (!in_array($typeExist->type, $newTypes)) {
        array_push($typesToDelete, TypesModel::GetIdTypeByName($typeExist->type));
    }
}
foreach ($typesToDelete as $typeId) {
    TypesModel::DeleteTypeOfGame($idGame, $typeId->id);
}

GamesModel::UpdateGame($date, $title, $description, $fileName,
    $platform, $idGame);

    header("Location: gestionJeux");
    exit;
} else {
    $message = "Veuillez sélectionner au moins un type.";
}
}
}
require "../src/Views/updateGameView.php";
}

/**
 * Modifier l'utilisateur
 *
 * @return void
 */
public function UpdateUser()
{
    $idUser = $_SESSION['idUser'];

    $btnSubmit = filter_input(INPUT_POST, "btnUpdateUser");
    $keepAvatar = filter_input(INPUT_POST, "keepAvatar");

    $message = "";

    if (isset($btnSubmit)) {
        $user = UsersModel::GetUserById($idUser);

        $avatar = $_FILES['file'];
        $old_pwd = filter_input(INPUT_POST, "oldPwd");

        if ($old_pwd != "") {
```

```
        if (password_verify($old_pwd, $user->password)) {

            $new_pwd = filter_input(INPUT_POST, "newPwd");
            $new_pwd_confirm = filter_input(INPUT_POST, "newPwdConfirm");

            if ($new_pwd === $new_pwd_confirm) {
                UsersModel::ChangePassword(password_hash($new_pwd,
PASSWORD_DEFAULT), $idUser);
            } else {
                $message = "Les nouveaux mot de passes ne sont pas
identiques.";
            }
        } else {
            $message = "L'ancien mot de passe est incorrect.";
        }
    }

    $old_avatar = $user->avatar;

    if ($avatar['name'] != "") {

        if ($old_avatar != null) {
            unlink("assets/images/" . $old_avatar);
        }

        $new_avatar = $avatar['name'];
        $file_tmp = $avatar['tmp_name'];

        $new_avatar = UsersModel::RenameAvatar($new_avatar, $user->login);
        move_uploaded_file($file_tmp, "assets/images/" . $new_avatar);

        UsersModel::ChangeAvatar($new_avatar, $idUser);
    } else if (!$keepAvatar) {

        if ($old_avatar != null) {
            unlink("assets/images/" . $old_avatar);
        }

        $new_avatar = null;

        UsersModel::ChangeAvatar($new_avatar, $idUser);
    }

    if ($message == "") {
        header("Location: profil");
    }
}
```



```
        exit;
    }
}

require "../src/Views/updateProfileView.php";
}
}
```

### LogoutController.php

```
<?php
/**
 * Contrôleur qui permet la déconnexion d'un utilisateur
 *
 * Billegas Lucas
 * 27.04.2023
 */

namespace ShareGames\Controllers;

class LogoutController
{
    /**
     * Déconnexion
     *
     * @return void
     */
    public function Logout()
    {
        // Vide la session
        $_SESSION = array();

        // Détruit la session
        session_destroy();

        header("Location: /");
        exit;
    }
}
```

### DeleteController.php

```
<?php

/**
```

```
* Contrôleur qui gère la suppression
*
* Billegas Lucas
* 03.05.2023
*/

namespace ShareGames\Controllers;

use ShareGames\Models\GamesModel;
use ShareGames\Models\OpinionsModel;

class DeleteController
{
    /**
     * Supprimer un avis
     *
     * @return void
     */
    public function DeleteOpinion()
    {
        $idOpinion = filter_input(INPUT_GET, "idOpinion");
        $idGame = filter_input(INPUT_GET, "idGame");

        OpinionsModel::DeleteOpinion($idOpinion);

        header("Location: detailsJeu?idGame=" . $idGame);
        exit;
    }

    /**
     * Supprimer d'un jeu
     *
     * @return void
     */
    public function DeleteGame(){
        if ($_SESSION['admin']) {
            $idGame = filter_input(INPUT_GET, "idGame");
            $game = GamesModel::GetGameById($idGame);

            unlink("assets/images/" . $game->vignette);
            GamesModel::DeleteGame($idGame);
        }

        header("Location: gestionJeux");
        exit;
    }
}
```

```
}
```

## Views

header.php

```
<div class="site-mobile-menu site-navbar-target">
    <div class="site-mobile-menu-header">
        <div class="site-mobile-menu-close">
            <span class="icofont-close js-menu-toggle"></span>
        </div>
    </div>
    <div class="site-mobile-menu-body"></div>
</div>
<nav class="site-nav">
    <div class="container">
        <div class="menu-bg-wrap">
            <div class="site-navigation">
                <div class="row g-0 align-items-center">
                    <div class="col-2">
                        <a href="/" class="logo m-0 float-start">ShareGames</a>
                    </div>
                    <div class="col-8 text-center">
                        <ul class="js-clone-nav d-none d-lg-inline-block text-start
site-menu mx-auto">

                            <li class="active"><a href="/">Home</a></li>
                            <li><a href="/jeux">Jeux</a></li>

                            <?php if (isset($_SESSION['userIsConnected'])) &&
$_SESSION['userIsConnected']) {
                                if($_SESSION['admin']){
                                    echo "<li><a href='/gestionJeux'>Gestion</a></li>";
                                }
                                echo "<li><a href='/profil'>Profil</a></li>";
                                echo "<li><a href='/deconnexion'>Deconnexion</a></li>";
                            } else {
                                ?>
                                <li><a href="/connexion">Connexion</a></li>
                                <li><a href="/inscription">Inscription</a></li>
                            <?php }?>

                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </div>
</nav>
```

```

        <div class="col-2 text-end">
            <a href="#" class="burger ms-auto float-end site-menu-toggle
js-menu-toggle d-inline-block d-lg-none light">
                <span></span>
            </a>
        </div>
    </div>
</div>
</div>
</div>
</nav>

```

footer.php

```

<footer class="site-footer">
    <div class="container">
        <div class="row">
            <div class="col-lg-4">
                <div class="widget">
                    <h3 class="mb-4">Contact</h3>
                    <p>Email: lucas.bllgs@eduge.ch</p>
                </div>
            </div>
            <div class="col-lg-4">
                <div class="widget">
                </div>
            </div>
        </div>
        <div class="row mt-5">
            <div class="col-12 text-center">
                <!--
                **=====
                NOTE:
                Please don't remove this copyright link unless you buy the license here
                https://untree.co/license/
                **=====
                -->
            </div>
        </div>
    </div>
</footer>

```

```
        <p>Copyright &copy;<script>document.write(new
Date().getFullYear());</script>. All Rights Reserved. &mdash; Designed with love by <a
href="https://untree.co">Untree.co</a> Distributed by <a
href="https://themewagon.com">Themewagon</a> License information:
https://untree.co/license/
    </p>
</div>
</div>
</div> <!-- /.container -->
</footer> <!-- /.site-footer -->

<script src="assets/js/bootstrap.bundle.min.js"></script>
<script src="assets/js/tiny-slider.js"></script>

<script src="assets/js/flatpickr.min.js"></script>

<script src="assets/js/aos.js"></script>
<script src="assets/js/glightbox.min.js"></script>
<script src="assets/js/navbar.js"></script>
<script src="assets/js/counter.js"></script>
<script src="assets/js/custom.js"></script>
```

## homeVlew.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">

    <link rel="shortcut icon" href="assets/images/favicon.png">

    <meta name="description" content="" />
    <meta name="keywords" content="bootstrap, bootstrap5" />

    <link rel="stylesheet" href="assets/fonts/icomoon/style.css">
    <link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">
```

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="assets/css/style.css">

<title>Accueil</title>
</head>

<body>
  <?php include "inc/header.php" ?>

  <section class="section">
    <div class="container">
      <div class="row mb-4">
        <div class="col-sm-6">
          <h2 class="posts-entry-title">Les 10 meilleurs jeux</h2>
        </div>
      </div>

      <!-- integration des cartes -->

      <?php echo $cardsTopGames; ?>

    </div>
  </section>

  <div class="section bg-light">
    <div class="container">

      <div class="row mb-4">
        <div class="col-sm-6">
          <h2 class="posts-entry-title">Les jeux les plus récents</h2>
        </div>
      </div>

      <?php echo $cardsMoreRecently; ?>

    </div>
  </div>
  <?php include "inc/footer.php"; ?>
</body>

</html>
```

## loginView.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">
    <link rel="shortcut icon" href="assets/images/favicon.png">

    <meta name="description" content="" />
    <meta name="keywords" content="bootstrap, bootstrap5" />

    <link rel="stylesheet" href="assets/fonts/icomoon/style.css">
    <link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">

    <link rel="stylesheet" href="assets/css/form.css">

    <link rel="stylesheet" href="assets/css/style.css">

    <title>Connexion</title>
</head>

<body>
    <?php include "inc/header.php"; ?>
    <form method="post" id="form">

        <label for="name">Nom: *</label>
        <input type="text" id="name" name="name" placeholder="Entrez votre nom"
required>

        <label for="password">Mot de passe: *</label>
        <input type="password" id="password" name="password" placeholder="Entrez votre
mot de passe" minlength="8" required>
```

```
<input type="submit" value="Se connecter" name="login">
<div style="margin-top: 1%;">
    <p class="small mb-3 pb-lg-1" style="color: red; text-align:center;"><?php
echo $message; ?></p>
</div>
<div style="text-align: center;">
    <p class="mb-0"><a href="" style="color: blue;"
onclick="ForgotPassword()">Mot de passe oublié ?</a> (Remplissez le champs nom avant de
cliquer)</p>
</div>
</form>
<?php include "inc/footer.php"; ?>
</body>
<script>
function ForgotPassword() {
    var form = document.getElementById('form');
    var formData = new FormData(form);

    var xhr = new XMLHttpRequest();

    xhr.open('POST', '/mdpOublier');
    xhr.send(formData);

    xhr.addEventListener('load', function() {
        if (xhr.status === 200) {
            alert(xhr.responseText);
        } else {
            console.error('Erreur de traitement du serveur');
        }
    });
}
</script>
</html>
```

#### registerView.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!DOCTYPE html>
<html lang="en">
```



```
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="author" content="Untree.co">
  <link rel="shortcut icon" href="assets/images/favicon.png">

  <meta name="description" content="" />
  <meta name="keywords" content="bootstrap, bootstrap5" />

  <link rel="stylesheet" href="assets/fonts/icomoon/style.css">
  <link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">

  <link rel="stylesheet" href="assets/css/form.css">
  <link rel="stylesheet" href="assets/css/style.css">

  <title>Inscription</title>
</head>

<body>
  <?php include "inc/header.php"; ?>
  <form method="post" enctype="multipart/form-data">

    <label for="name">Nom: *</label>
    <input type="text" id="name" name="name" placeholder="Entrez votre nom"
required>

    <label for="password">Mot de passe: *</label>
    <input type="password" id="password" name="password" placeholder="Entrez votre
mot de passe" minlength="8" required>

    <label for="password">Confirmation du mot de passe: *</label>
    <input type="password" id="confirmPassword" name="confirmPassword"
placeholder="Confirmez votre mot de passe" minlength="8" required>

    <label for="avatar">Sélectionnez un avatar (optionnel) :</label>
    <input type="file" name="avatar" accept="image/*">

    <input type="submit" value="S'inscrire" name="register">
    <div style="margin-top: 1%;>
      <p class="small mb-3 pb-lg-1" style="color: red; text-align:center;"><?php
echo $message; ?></p>
    </div>
  </div>
```

```
        <p class="mb-0" style="text-align:center;">Vous avez déjà un compte ? <a
href="/connexion">Connexion</a></p>
    </div>
</form>
<?php include "inc/footer.php"; ?>
</body>

</html>
```

### profileView.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">
    <link rel="shortcut icon" href="assets/images/favicon.png">

    <meta name="description" content="" />
    <meta name="keywords" content="bootstrap, bootstrap5" />

    <link rel="stylesheet" href="assets/css/style.css">

    <title>Profil</title>
</head>

<body>
    <?php include "inc/header.php"; ?>

    <section class="h-100">
        <div class="container py-5 h-100">
            <div class="row d-flex justify-content-center align-items-center h-100">
                <div class="col col-lg-9 col-xl-7">
                    <div class="card">
                        <div class="rounded-top text-white d-flex flex-row"
style="background-color: darkblue; height:200px;">
                            <div class="ms-4 mt-5 d-flex flex-column" style="width:
150px;">
```

```

        
        <a href="modifierProfil" data-mdb-ripple-color="dark"
style="z-index: 1;" class="btn btn-outline-primary">Modifier profil</a>
    </div>
    <div class="ms-3" style="margin-top: 130px;">
        <h5><?= $user->login ?></h5>
    </div>
</div>
<div class="p-4 text-black" style="background-color: #f8f9fa;">
    <div class="d-flex justify-content-end text-center py-1">

        <div>
            <p class="mb-1 h5"><?= $nbCommentary ?></p>
            <p class="small text-muted mb-0">Commentaires</p>
        </div>
    </div>
</div>
<div class="card-body p-4 text-black">

    <div class="d-flex justify-content-between align-items-
center mb-4">

        <p class="lead fw-normal mb-0">Les jeux que vous avez
commentés</p>

    </div>

    <?php echo $displayGames; ?>
</div>
</div>
</div>
</div>
</div>
</section>

    <?php include "inc/footer.php"; ?>
</body>

</html>

```

gamesView.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
```

```
*/ -->
<!Doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="author" content="Untree.co">
  <link rel="shortcut icon" href="assets/images/favicon.png">

  <meta name="description" content="" />
  <meta name="keywords" content="bootstrap, bootstrap5" />

  <link rel="stylesheet" href="assets/fonts/icomoon/style.css">
  <link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

  <link rel="stylesheet" href="assets/css/style.css">

  <title>Jeux</title>
</head>

<body>
  <?php include "inc/header.php"; ?>
  <section class="section">

    <div class="container" style="margin-bottom: 3%;">
      <form method="post" style="margin-left: 35%; margin-right: 35%; width: 30%;
margin-bottom: 2%;">
        <div style="flex-direction:row; display:flex;">
          <div class="col-6 mb-3">
            <select name="filter" class="form-control">
              <option value="">Aucun</option>
              <option value="date">Date</option>
              <option value="mark">Note</option>
            </select>
          </div>
          <div class="col-12">
            <input type="submit" name="btnFilter" value="Valider"
class="btn btn-outline-primary card">
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```
        </form>
        <form method="post" style="margin-left: 35%; margin-right: 35%; width:
30%;">
            <div style="display: flex;">
                <input type="text" name="search" class="form-control"
placeholder="Rechercher...">
                <button type="submit" name="btnSearch" class="btn btn-outline-
primary"><span class="bi-search"></span></button>
            </div>
        </form>
        <br>
        <p style="text-align: center; color:red;"><?=$messageError?></p>
    </div>
    <div class="container">
        <div class="row">
            <?php echo $displayGames; ?>
        </div>
    </div>
</section>

<?=$displayPagination?>

    <?php include "inc/footer.php"; ?>
</body>

</html>
```

## detailsGameVlew.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">
    <link rel="shortcut icon" href="assets/images/favicon.png">

    <meta name="description" content="" />
```

```
<meta name="keywords" content="bootstrap, bootstrap5" />

<link rel="stylesheet" href="assets/fonts/icomoon/style.css">
<link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">

<link rel="stylesheet" href="assets/css/style.css">

<title>Détails du jeu</title>
</head>

<body>
  <?php include "inc/header.php"; ?>

  <div class="post-entry text-center">
    <h1 class="mb-4" style="color: black; margin-top: 1%;"><?= $game->titre ?></h1>
    <div class="post-meta align-items-center text-center">
      <span style="font-size: 18px;"><?= date("d-m-Y", strtotime($game->date));
?></span>
    </div>
  </div>

  <section class="section">
    <div class="container">
      <div class="row blog-entries element-animate">
        <div class="col-md-12 col-lg-8 main-content">
          <div class="post-content-body">
            <div class="row my-4">
              <div class="col-md-12 mb-4">
                <?php
                  echo "<img src='assets/images/' . $game->vignette . ' ' alt='image'
style='width:70%; margin-left:30%;'>";
                ?>
              </div>
            </div>
          </div>
          <div class="pt-5 comment-wrap">
            <h3 class="mb-5 heading"><?= $nbOpinions ?> Commentaires</h3>
            <?php echo $displayOpinions; ?>
            <div class="comment-form-wrap pt-5">
              <h3 class="mb-5">Commenter le jeu</h3>
              <form method="post" class="p-5 bg-light">
                <div class="form-group">
                  <label for="title">Titre *</label>
```

```

        <input type="text" class="form-control" id="title" name="title"
required>
    </div>
    <div class="form-group">
        <label for="mark">Note *</label>
        <input type="number" min="0" max="10" class="form-control" id="mark"
name="mark" required>
    </div>

    <div class="form-group">
        <label for="description">Description</label>
        <textarea name="description" id="description" cols="30" rows="10"
class="form-control"></textarea>
    </div>
    <div class="form-group">
        <?php if(!isset($_SESSION['userIsConnected'])) ||
!$_SESSION['userIsConnected']>{ ?>
            <input type="submit" value="Poster un commentaire"
name="btnSubmitOpinion" class='btn btn-sm btn-outline-primary' style="margin-right:25%;
margin-left:25%; width: 50%; disabled>
            <p style="text-align: center; color:red">Connectez-vous pour
commenter le jeu</p>
            <?php }else{ ?>
                <input type="submit" value="Poster un commentaire"
name="btnSubmitOpinion" class='btn btn-sm btn-outline-primary' style="margin-right:25%;
margin-left:25%; width: 50%;">
                <?php } ?>
            </div>
        </form>
    </div>
</div>
<div class="col-md-12 col-lg-4 sidebar">
    <div class="sidebar-box">
        <h3 class="heading">Type du jeu</h3>
        <ul class="tags">
            <?php echo $listTypes; ?>
        </ul>
        <br><br>
        <h3 class="heading">Plateforme</h3>
        <ul class="tags">
            <li><a href=""><?= $game->plateforme ?></a></li>
        </ul>
        <br><br>
        <h3 class="heading">Description</h3>
        <p><?= $game->description ?></p>

```

```
        </div>
    </div>
</div>
</div>
</section>

<?php include "inc/footer.php"; ?>
</body>

</html>
```

### createGameVlew.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">
    <link rel="shortcut icon" href="assets/images/favicon.png">

    <meta name="description" content="" />
    <meta name="keywords" content="bootstrap, bootstrap5" />

    <link rel="stylesheet" href="assets/fonts/icomoon/style.css">
    <link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">

    <link rel="stylesheet" href="assets/css/form.css">

    <link rel="stylesheet" href="assets/css/style.css">

    <title>Création d'une fiche de jeu</title>
</head>

<body>
```



```
<?php include "inc/header.php"; ?>
<form method="post" enctype="multipart/form-data">

    <label for="title">Titre: *</label>
    <input type="text" id="title" name="title" placeholder="Entrez le titre"
required>

    <label for="description">Description (optionnel) :</label>
    <textarea name="description" rows="10" cols="40"></textarea>
    <br>
    <label for="platform">Sélectionner la plateforme:</label>
    <select name="platform" id="platform">
        <option value="PS">PS</option>
        <option value="XBOX">XBOX</option>
        <option value="PC">PC</option>
        <option value="Mobile">Mobile</option>
        <option value="autre" selected>Autre</option>
    </select>
    <br>
    <label for="types">Sélectionner les types du jeu: *</label>

    <?php echo $checkbox; ?>

    <br>
    <label for="vignette">Sélectionnez une vignette : *</label>
    <input type="file" name="file" accept="image/*">

    <input type="submit" value="Création de la fiche" name="btnCreate">
    <p style="color:red; text-align:center;"><?php echo $message; ?><p>
</form>
<?php include "inc/footer.php"; ?>
</body>

</html>
```

## managementGamesView.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">
    <link rel="shortcut icon" href="assets/images/favicon.jpg">

    <meta name="description" content="" />
    <meta name="keywords" content="bootstrap, bootstrap5" />

    <link rel="stylesheet" href="assets/fonts/icomoon/style.css">
    <link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" href="assets/css/style.css">

    <title>Gestion des jeux</title>
</head>

<body>
    <?php include "inc/header.php"; ?>

    <section class="section">
        <h1 class="heading text-black mb-3" style="text-align: center; margin-top: 2%;
data-aos="fade-up">Gestion des jeux</h1>
        <br>

        <div class="container" style="margin-bottom: 3%;">
            <a href="/creationJeu" class='btn btn-outline-primary'>Cr  rer une fiche de
jeu</a>
        </div>
        <div class="container">
            <div class="row">
```

```
        <?php echo $displayGames; ?>
    </div>
</div>
</section>

<?= $displayPagination ?>

<?php include "inc/footer.php"; ?>
</body>

</html>
```

### updateGameVlew.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">
    <link rel="shortcut icon" href="assets/images/favicon.png">

    <meta name="description" content="" />
    <meta name="keywords" content="bootstrap, bootstrap5" />

    <link rel="stylesheet" href="assets/fonts/icomoon/style.css">
    <link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">

    <link rel="stylesheet" href="assets/css/form.css">

    <link rel="stylesheet" href="assets/css/style.css">

    <title>Modification d'une fiche de jeu</title>
</head>

<body>
```

```
<?php include "inc/header.php"; ?>
<div>
    <form method="post" enctype="multipart/form-data">
        <div class="row">
            <div class="col-6 mb-3">
                <label for="title">Titre: *</label>
                <input type="text" id="title" name="title" placeholder="Entrez le
titre" value="<?= $title ?>" required>
            </div>
            <div class="col-12 mb-3">
                <label for="platform">Sélectionner la plateforme: *</label>
                <select name="platform" id="platform">
                    <option value="PS">PS</option>
                    <option value="XBOX">XBOX</option>
                    <option value="PC">PC</option>
                    <option value="Mobile">Mobile</option>
                    <option value="autre" selected>Autre</option>
                </select>
            </div>
            <div class="col-12 mb-3">
                <label for="types">Sélectionner les types du jeu: *</label>
                <?php echo $checkboxes; ?>
            </div>
            <div class="col-12 mb-3">
                <label for="vignette">Sélectionnez une vignette : *</label>
                <input type="file" name="file" accept="image/*">
            </div>
            <div class="col-12 mb-3">
                <label for="description">Description (optionnel) :</label>
                <textarea name="description" cols="30" rows="7" class="form-
control"><?= $description ?></textarea>
            </div>

            <input type="submit" value="Modification de la fiche"
name="btnUpdateGame">

            <p style="color:red; text-align:center;"><?php echo $message; ?>
            <p>
        </div>
    </form>
</div>

<?php include "inc/footer.php"; ?>
</body>
</html>
```

updateOpinionView.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">
    <link rel="shortcut icon" href="assets/images/favicon.png">

    <meta name="description" content="" />
    <meta name="keywords" content="bootstrap, bootstrap5" />

    <link rel="stylesheet" href="assets/fonts/icomoon/style.css">
    <link rel="stylesheet" href="assets/fonts/flaticon/font/flaticon.css">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.8.1/font/bootstrap-icons.css">

    <link rel="stylesheet" href="assets/css/style.css">
    <link rel="stylesheet" href="assets/css/form.css">

    <title>Modification d'un avis</title>
</head>

<body>
    <?php include "inc/header.php"; ?>
    <div class="pt-5 comment-wrap">
        <div class="comment-form-wrap pt-5">
            <h3 style="text-align: center;">Modifier le commentaire</h3>
            <form method="post" class="p-5 bg-light">
                <div class="row">
                    <div class="col-12 mb-3">
                        <label for="title">Titre *</label>
                        <input type="text" class="form-control" id="title" name="title"
value="<?= $title ?>" required>
                    </div>
                    <div class="col-12 mb-3">
                        <label for="mark">Note *</label>
```

```
        <input type="number" min="0" max="10" class="form-control"
id="mark" name="mark" value="<?= $mark ?>" required>
    </div>

    <div class="col-12 mb-3">
        <label for="description">Description</label>
        <textarea name="description" id="description" cols="30"
rows="10" class="form-control"><?= $description ?></textarea>
    </div>

    <input type="submit" class="submitOpinion" value="Modifier le
commentaire" class="btn btn-outline-primary" name="btnSubmitOpinion">
    </form>
</div>
</div>
<?php include "inc/footer.php"; ?>
</body>

</html>
```

#### updateProfileView.php

```
<!-- /*
* Template Name: Blogy
* Template Author: Untree.co
* Template URI: https://untree.co/
* License: https://creativecommons.org/licenses/by/3.0/
*/ -->
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="author" content="Untree.co">
    <link rel="shortcut icon" href="assets/images/favicon.png">

    <meta name="description" content="" />
    <meta name="keywords" content="bootstrap, bootstrap5" />

    <link rel="stylesheet" href="assets/fonts/icomoon/style.css">

    <link rel="stylesheet" href="assets/css/form.css">

    <link rel="stylesheet" href="assets/css/style.css">
```

```
<title>Modification du profil</title>
</head>

<body>
  <?php include "inc/header.php"; ?>
  <div>
    <form method="post" enctype="multipart/form-data">
      <div class="row">
        <div class="col-12 mb-3">
          <label for="oldPassword">Ancien mot de passe:</label>
          <input type="password" id="oldPwd" name="oldPwd" minlength="8"
placeholder="Entrez votre ancien mot de passe">
        </div>
        <div class="col-6 mb-3">
          <label for="newPassword">Nouveau mot de passe:</label>
          <input type="password" id="newPwd" name="newPwd" minlength="8"
placeholder="Entrez votre nouveau mot de passe">
        </div>
        <div class="col-6 mb-3">
          <label for="newPasswordConfirm">Confirmation nouveau mot de
passe:</label>
          <input type="password" id="newPwdConfirm" name="newPwdConfirm"
minlength="8" placeholder="Confirmez votre nouveau mot de passe">
        </div>

        <div class="col-12 mb-3">
          <label for="avatar">Sélectionnez un nouveau avatar:</label>
          <input type="file" name="file" accept="image/*">

          <input type="checkbox" name="keepAvatar" id="keepAvatar" checked>
Garder votre avatar
        </div>

        <div class="col-12">
          <input type="submit" value="Modification de votre profil"
name="btnUpdateUser">
        </div>
        <p style="color:red; text-align:center;"><?php echo $message; ?>
        <p>
      </div>
    </form>
  </div>

  <?php include "inc/footer.php"; ?>
</body>
```

</html>