

DAT Projet Web

*Indé-Couvertes, site de découverte
de jeux vidéo indépendants*

Johanna BOITEUX



Sommaire

I. Introduction.....	3
A. Contexte.....	3
B. Fonctionnalités.....	3
II. Gestion des données.....	4
A. Règles de gestion.....	4
B. Schéma de la base de données.....	6
III. Implémentation du site.....	7
A. Schéma de déploiement.....	7
B. Architecture de l'application.....	8
IV. Justification des choix.....	9
A. Langages et logiciels.....	9
B. Forme d'hébergement.....	10
V. Post-Mortem.....	12
A. Ce qui s'est bien passé.....	12
B. Ce qui s'est mal passé.....	12
C. Suggestion d'amélioration.....	12

I. Introduction

A. Contexte

De nos jours, l'industrie du jeu vidéo est dominée par quelques grands éditeurs à l'influence considérable ; des noms tels qu'Ubisoft, Blizzard, Activision ou Electronic Arts sont en effet immédiatement reconnus comme des poids lourds même par les non joueurs. Ces mastodontes du jeu vidéo monopolisent le marché en lançant, souvent en période de fêtes, des titres phares qui se vendent à des millions d'exemplaires, d'où leur étiquette de « Triple A ».

S'ils se distinguent par leurs ventes colossales, ces jeux « Triple A » ne brillent pas, en revanche, par leur originalité : suites de licences à succès, surexploitation des mêmes genres, direction artistique oubliable, gameplay uniformisé, narration peu voire pas présente. Il y a donc malheureusement un manque de prises de risques et d'innovation chez ces grands éditeurs.

Ce serait pourtant ignorer l'immense champ des possibles permis par un média créatif aussi libre et riche que celui des jeux vidéo ; champ qu'explorent aujourd'hui les petits éditeurs de jeux. Ne dépendant d'aucune maison mère et d'aucun investisseur privé, ces multitudes de petits studios émergents sont donc qualifiés d'« indépendants », terme qui s'applique aussi à leurs productions. Plutôt méconnus du grand public, ces « jeux indés » offrent pourtant aux joueurs des expériences inédites, parfois étranges, certaines inoubliables, mais toujours innovantes.

Hélas, le manque de moyens des petits éditeurs empêche de faire connaître leurs créations innovantes à un public qui commence à se lasser des « Triple A ». Sans compter que les ventes modestes de leurs productions condamnent trop souvent des développeurs créatifs et talentueux à fermer leurs studios. Une injustice que le site Indé-Couvertes compte bien abolir !

B. Fonctionnalités

Indé-Couvertes est le site incontournable de découverte des jeux indés qui vous correspondent. Il s'adresse certes aux joueurs aguerris, mais cherche principalement à faire découvrir aux joueurs occasionnels – et surtout aux non joueurs ! – l'univers foisonnant des jeux indés. Si ce site fait naître chez vous une passion pour les jeux indés, alors il aura atteint son objectif !

Vous pouvez accéder à Indé-Couvertes en tant que simple visiteur pour rechercher dans notre base de données (qui se veut la plus complète possible grâce à des mises à jour régulières) les titres indés qui correspondent à vos goûts, que ce soit votre éditeur préféré, votre plate-forme favorite ou votre genre de prédilection !

De plus, l'espace Premium offre à ses membres la possibilité de contribuer au développement de notre base de données en ajoutant, modifiant et supprimant nos jeux. Vous pourrez également noter et commenter nos jeux, mais aussi voir les jeux consultés par vos amis et recevoir des suggestions de jeux les plus populaires. En vous inscrivant à notre espace Premium, vous bénéficierez en outre d'offres de réduction sur les jeux les mieux notés grâce à nos partenariats avec de nombreux studios indépendants.

Des milliers de perles indés sont désormais à votre portée, alors n'attendez plus !

II. Gestion des données

A. Règles de gestion

A la page d'accueil, on demande à l'internaute s'il souhaite accéder au site en tant que simple visiteur ou en membre Premium : ces deux types d'internautes seront par la suite respectivement désignés par A et B. On distribuera donc les fonctionnalités du site en fonction du type d'internaute.

1) Inscription et identification

Type A : Pas d'inscription ni d'identification requise. L'internaute accède directement à la page de Recherche standard (sans les options réservées au Type B).

Type B : A la première inscription, l'internaute doit renseigner son Prénom et son Nom, son AdresseMail, son MotDePasse et son Pseudo, qui lui servira d'identifiant sur le site.

Pour afficher les jeux que les amis du membre ont consulté, on associera à chaque membre une liste d'amis (qui peut être vide) eux aussi membres Premium.

A l'identification, il devra simplement donner son Pseudo et son MotDePasse. Deux comptes Premium distincts ont donc des Pseudo et des MotDePasse différents. En revanche, ils peuvent avoir en commun leur Nom et Prénom (en cas d'homonymie), leur AdresseMail (un même internaute peut posséder plusieurs comptes Premium) et certains amis.

2) Recherche

Types A et B : Chaque internaute peut lancer au moins une Recherche de jeux sur la page dédiée. Une Recherche, identifiée par un NumRecherche, comporte un booléen ATrouvé vérifiant si elle renvoyé au moins un résultat.

Chaque Recherche peut ou non comporter au moins un Critère (note, genre, éditeur, année de sortie, mot-clé, etc) permettant d'affiner et de grouper les résultats. Plusieurs Critères peuvent se cumuler lors d'une même Recherche.

Afin d'optimiser l'algorithme de la Recherche, on stocke sa Durée et une NotePertinence donnée par les internautes à l'issue de celle-ci.

Type B uniquement : Pour le journal des activités récentes du membre Premium, on conservera son Pseudo et le(s) NumRecherche ainsi que les IdJeu des Jeu consultés.

3) Jeu

Types A et B : Chaque Jeu consulté est identifié par un IdJeu, et comporte un Titre, une Image de jaquette, une Description, un Prix, une AnnéeDeSortie, un Editeur, un Genre, une Plate-Forme ainsi qu'une liste de mots-clés pouvant servir lors d'une Recherche. Il comporte aussi un NbConsultations permettant de hiérarchiser les résultats d'une Recherche ; suivant le Critère de la Recherche, ces jeux seront classés par NbConsultations décroissant.

Type B uniquement : Pour les suggestions des jeux les plus consultés, on gardera les IdJeu des dix Jeu ayant les NbConsultations les plus élevés.

Facultatif : Pour les suggestions de jeux susceptibles de plaire au membre Premium, on conservera en plus le Critère de chaque Recherche qu'il a effectuée.

4) **Ajout/Suppression/Modification des jeux**

Type B : Un membre Premium peut ajouter et modifier autant de Jeu qu'il le souhaite, mais ne peut supprimer que trois Jeu de la base de données.

N.B. : L'Evaluation d'un Jeu n'est pas considérée comme une modification.

5) **Evaluation**

Type B : Chaque membre Premium peut évaluer plusieurs Jeu, et un même Jeu peut avoir plusieurs Evaluation de membres distincts (deux Evaluations sur un même Jeu ne peuvent donc pas être faites sous le même Pseudo).

Une Evaluation, identifiée par une IdEval, comporte une Note entière entre 0 et 20. Cette Note peut ou non être accompagnée d'un Commentaire. Lorsqu'un Jeu est évalué au moins une fois, il reçoit une NoteMoyenne calculée à l'aide de toutes les notes données par les membres Premium.

6) **Facultatif : Offre de réduction**

Type B : Par défaut, aucun membre Premium n'a droit à une réduction de la part d'un éditeur Partenaire. Une Réduction, identifiée par le Pseudo du membre Premium bénéficiaire et l'IdEditeur de l'éditeur partenaire, contient donc un MontantReduc positif initialisé à 0 et une liste d'Editeur partenaires vide (Un Editeur n'est pas forcément partenaire).

Chaque membre devra mériter ces remises (et augmenter MontantReduc) en donnant au moins 10 Evaluations complètes (Note et Commentaire) sur des Jeu produits par un même Editeur partenaire.

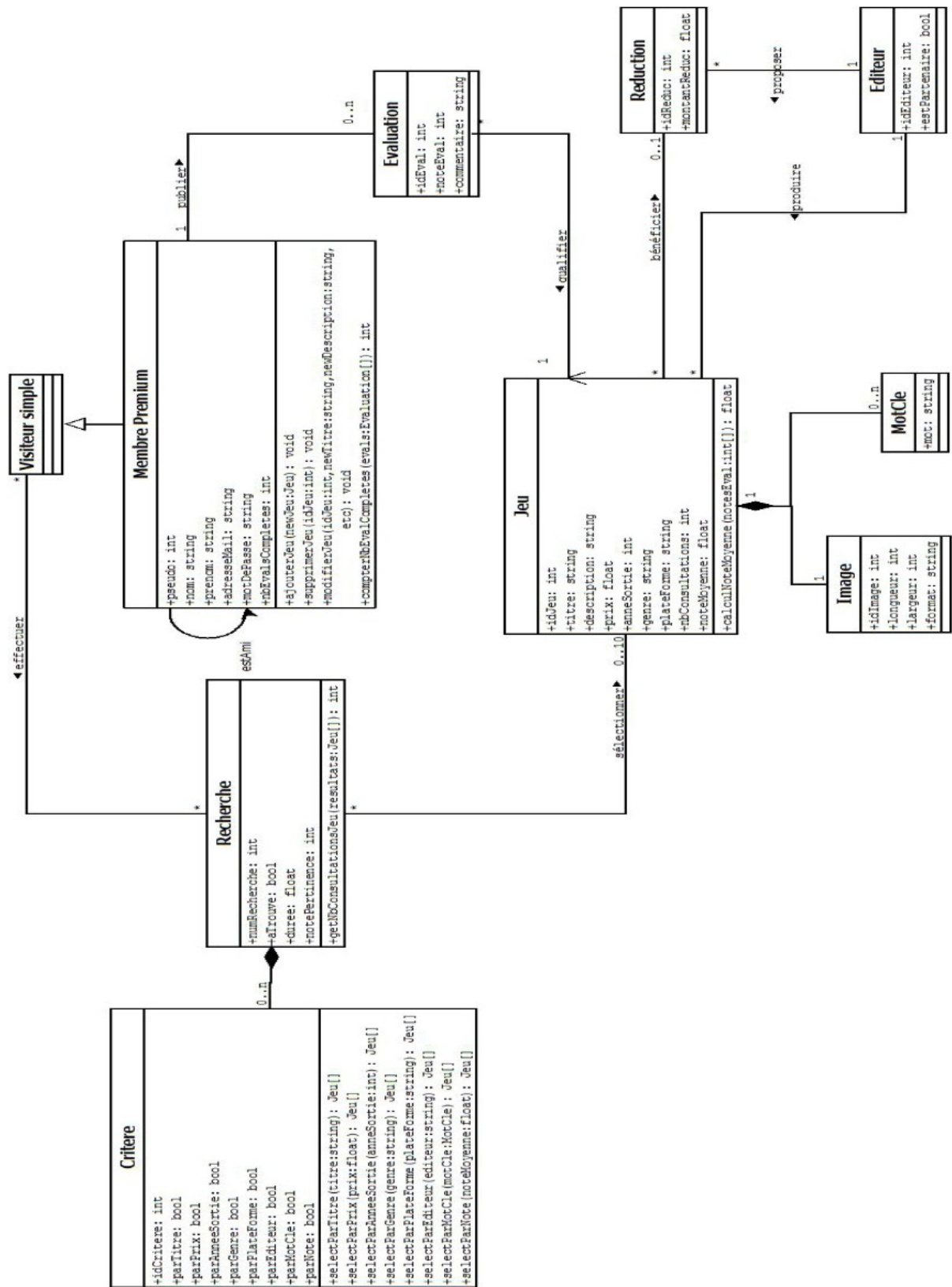
Un même membre Premium peut ainsi cumuler les Réduction, mais seulement chez des Editeur partenaires différents.

Pour les offres de réductions, on sélectionne chaque Jeu où MontantReduc>0. Puis, pour chaque Pseudo distinct ayant publié une Evaluation complète (Note et Commentaire) de ce Jeu, s'il a fait pareil pour au moins 9 autres Jeu du même Partenaire et dont le MontantReduc>0.

7)

B. Schéma de la base de données

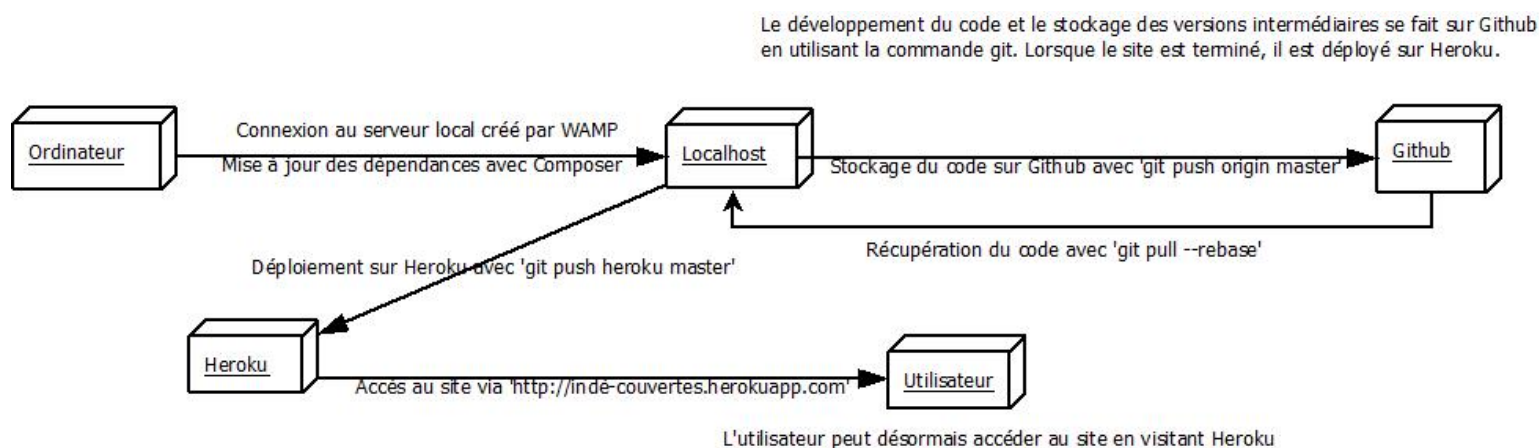
Diagramme UML de la base de données *JeuxIndés* réalisé avec le logiciel Dia :



III. Implémentation du site

A. Schéma de déploiement

Diagramme de déploiement du site web *Indé-Couvertes* réalisé avec le logiciel Dia :



Déploiement du site PHP en local avec WAMP et Composer

WAMP (Windows Apache MySQL PHP) est un logiciel permettant de créer le serveur et la base de données sur l'ordinateur. Il y crée un dossier WAMP, dont le chemin par défaut est C:\wamp. Ce dossier contient un répertoire www regroupant l'ensemble des fichiers qui seront gérés par le serveur web. C'est là que se trouve le dossier Projet Indes-Couvertes contenant le site.

Composer est un outil de gestion de dépendances en PHP permettant de les déclarer, de les installer et de les mettre à jour automatiquement. Les dépendances sont toutes les librairies dont le site dépend pour fonctionner. On définit dans un fichier composer.json (situé dans Projet Indes-Couvertes) la liste des librairies dont le site dépend, ainsi que leur version. On exécute ensuite une commande pour installer ou mettre à jour ces librairies (et leurs propres dépendances). Puis, on inclut le fichier d'autoload généré par Composer dans notre projet.

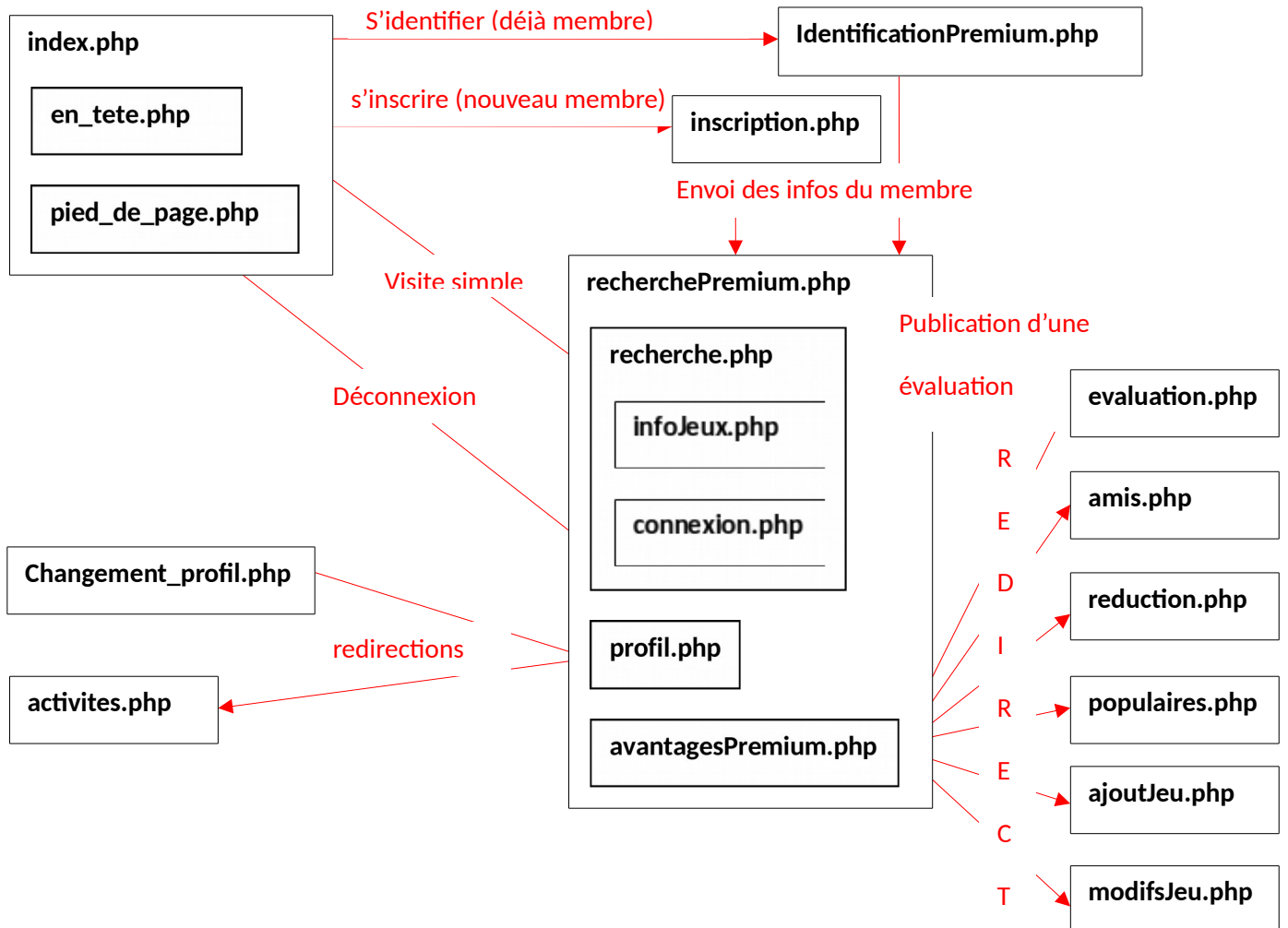
Composer détermine ensuite la meilleure version possible de ces dépendances, les télécharge, et configure automatiquement leur *autoload*. Composer est donc indiqué lorsqu'on utilise des librairies externes. En effet, ces librairies doivent être mises à jour une à une pour s'assurer de corriger les bogues de chacune d'entre elles. De plus, ces librairies peuvent elles-mêmes dépendre d'autres librairies, ce qui oblige à gérer l'ensemble de ces dépendances (installation, mises à jour, etc.). Enfin, ces librairies ont chacune leur paramètres d'*autoload*, et vous devez gérer leur *autoload* pour chacune d'entre elles. Composer trouve toutes les librairies sur le site <http://www.packagist.org> et peut alors télécharger les dépendances PHP.

Déploiement du site en PHP sur Heroku

Cette opération nécessite un [compte Heroku](#) + [PHP](#) et [Composer](#) installés en local

B. Architecture de l'application

Les fichiers en_tete.php et pied_de_page.php sont inclus dans tous les autres fichiers



IV. Justification des choix

A. Langages et logiciels

- **PHP** : Un langage disponible gratuitement que seuls les serveurs comprennent et qui permet de rendre un site dynamique. Il se démarque de ses concurrents par une importante communauté qui peut aider rapidement sur Internet en cas de problème. C'est un langage facile à utiliser, idéal pour les débutants comme pour les professionnels : Wikipédia et Facebook sont des exemples de sites célèbres et très fréquentés qui fonctionnent grâce à PHP.

PHP génère du HTML : les clients sont incapables de comprendre le code PHP, ne connaissant que le HTML et le CSS. Seul le serveur est capable de lire du PHP. Le rôle de PHP est justement de générer du code HTML, qui est ensuite envoyé au client.

PHP est aussi un langage de programmation utilisé sur de nombreux serveurs pour prendre des décisions. C'est toujours PHP qui décide du code HTML qui sera généré et envoyé au client. C'est donc PHP qui génère la page web.

PHP peut fonctionner seul et suffit à créer un site dynamique, mais offre de nombreuses possibilités lorsqu'on le combine à un SGBD comme MySQL. La combinaison **PHP+MySQL** est en effet la plus courante.

- **MySQL** : Un Système de Gestion de Base de Données (SGBD) permettant d'enregistrer des données de manière organisée afin de les retrouver facilement plus tard. C'est grâce à MySQL que l'on pourra enregistrer la liste des membres Premium du site ou leurs activités récentes. Le langage qui permet de communiquer avec la base de données s'appelle le SQL. MySQL est le plus indiqué si l'on a peu de moyens car il est libre, gratuit, performant et utilisé par de nombreuses personnes susceptibles d'aider.

PHP et MySQL sont des logiciels libres, ce qui garantit leur pérennité : par conséquent, tout le monde peut contribuer à leur développement.

- **phpMyAdmin** : Une interface écrite en PHP pour gérer des bases de données MySQL. C'est un ensemble de pages PHP toutes prêtes permettant de gagner du temps.

Des URI propres dans les fichiers CSS évitent au navigateur internet de faire des requêtes http supplémentaires pour charger les ressources externes, ce qui augmente la vitesse de chargement de la page.

B. Forme d'hébergement

La Plateforme en tant que Service (PaaS), un service Cloud Computing permettant d'externaliser l'hébergement des outils logiciels et matériels de développement d'applications. La PaaS propose notamment un système de gestion de base de données et un logiciel de serveur.

Comme la PaaS prend en charge toute la partie informatique et logicielle, le hardware et le software sont donc hébergés sur l'infrastructure du fournisseur. Ainsi, on n'a pas besoin d'installer son propre hardware et ses logiciels en interne pour développer ou lancer de nouvelles applications.

De plus, la PaaS ne prend pas en compte l'aspect serveur. Il suffit donc de déployer, via un navigateur web, l'application fonctionnant avec l'un des langages supportés.

- **Heroku** : La plus connue des PaaS publiques de développement et de gestion de logiciel. L'architecture Heroku est centrée sur les processus, et non sur des serveurs. Le code déployé est analysé pour détecter le type d'application à faire tourner ; l'analyse déclenche la mise en place d'un environnement dédié, adossé à une (des) unité(s) d'exécution appelées « dynos ». Ces dynos gèrent les processus faisant fonctionner l'application déployée. Heroku est capable de déclarer et scaler automatiquement ces processus, l'utilisateur pouvant affiner ses réglages par défaut. Un dyno contient un (des) processus web, un (des) worker, et un (des) processus système. L'ensemble des dynos requis pour faire tourner une application Heroku forment une unité matérielle logique appelée « slug ». Le slug est scalable horizontalement à l'infini en modifiant la physionomie des dynos, même si l'architecture des applications web empêche souvent un scaling trop agressif (en cas d'accès à une base de données lente). Le déploiement de code sur un slug, ou la modification de variables d'environnements, entraînent une recompilation du slug. Ainsi, le(s) dyno(s) et le matériel qui le(s) composent sont réadaptés aux nouvelles conditions d'exécution de l'application. Les dynos améliorent donc la scalabilité de l'application, tandis que les slug la rendent plus modulable. Heroku a également beaucoup d'avantages : elle permet un déploiement rapide d'applications web dans le Cloud, avec une gestion très souple du scaling horizontal grâce à un modèle de gestion des processus adapté au web. Elle supporte aussi de nombreuses applications et extensions, comme [ce buildpack PHP](#) qui permet d'étendre les langages supportés ou d'augmenter les capacités de l'environnement (mise en place de bases de données spécifiques). Elle possède d'autres points forts : pas de gestion de serveur ni de maintenance pour les mises à jour, et installation des services supplémentaires comme une base de données simplifiée. De plus, on ne paye que ce que l'on utilise comme puissance. En montant les dynos sur les horaires voulus, on ne paiera que la tranche horaire demandée.

L'environnement de déploiement et d'exécution de Heroku est accessible via un ensemble d'outils spécifiques (le heroku tool belts) et Git, à travers la ligne de commande. D'où le choix de Github comme service web d'hébergement et de gestion de développement.

- **Github** : Le plus grand hébergeur de code source au monde utilisant le logiciel de gestion de versions Git. Git garde une trace des modifications apportées à chaque fichier afin de pouvoir

facilement revenir en arrière ou à une version spécifique. Github assure un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, la gestion de tâches et un wiki pour chaque projet.

GitHub repose sur le principe du [fork](#) (embranchement) par défaut : toute personne « forkant » le projet devient alors publiquement le leader de son projet du même nom que l'original. Si on trouve un projet intéressant sur Github, on peut donc le forker pour devenir le propriétaire d'un projet similaire du même nom.

En plus d'offrir l'hébergement de projets avec [Git](#), le site possède de nombreuses fonctionnalités dignes d'un réseau social comme les flux, la possibilité de suivre des personnes ou des projets ainsi que des graphes de réseaux pour les [dépôts](#) (repository). Grâce à Github, on peut donc suivre de près l'évolution du nombre de projets et y prendre part. Il est également possible de créer gratuitement son propre projet open-source.

GitHub offre aussi un [logiciel de suivi de problèmes](#) (issue tracking system). D'autre part, les développeurs améliorent cette plateforme en permanence, ce qui leur sera également bénéfique.

V. Post-Mortem

A. Ce qui s'est bien passé

- Bonne compréhension du fonctionnement d'un site en php avec MySQL ainsi que du rôle de chacun des acteurs impliqués, qu'ils soient logiciels ou hébergeurs.
- Création et exportation de la base de données MySQL JeuxIndés.
- Dossiers de dépôt Github et Heroku créés et mis à jour avec succès.

B. Ce qui s'est mal passé

- Quelques soucis lors du téléchargement de certains logiciels comme Composer.
- Problèmes de conflits lors des git, qui bien que résolus ont considérablement ralenti l'avancée du projet.
- Manque de temps pour remplir suffisamment la base de données JeuxIndés.

C. Suggestion d'amélioration

- Mieux se renseigner sur les modalités de téléchargement et d'installation des logiciels pour éviter de perdre du temps en cas de problème.
- Passer du temps supplémentaire à expérimenter sur les commandes git pour mieux se familiariser avec leurs fonctionnalités.
- Opter pour le remplissage automatique de la base de données.