

Réunion entre groupes sur la communication d'informations plateforme - bibliothèque de jeux

8 décembre 2023

Participants

Nom	Prénom	Groupe
Binginot	Etienne	1
Mazuy	Axelle	1
Montagne	Erwann	1
Brahimi	Sid Ahmed	2
El Gharbaoui	Anass	2

Spectateurs

Nom	Prénom	Groupe
Cauchois	Niels	1
Ducroq	Yohann	1
Thiberville	Malvina	1
Van Liedekerke	Florian	1

Ordres du jour

- Discussion sur la communication, la "boîte noire" des deux projets
- Discussion sur l'agencement du Git
- Technologies utilisées et conventions

Table des matières

1	Discussion sur la communication, la "boîte noire" des deux projets	3
1.1	Service "GameService"	3
1.2	Historique des coups	3
1.3	Informations sur les jeux, configuration du nombre de joueurs . .	3
2	Discussion sur l'agencement du Git	3
2.1	Branche "développement"	3
2.2	Branche "pré-production"	4
2.3	Branche "production"	4
2.4	Commits et conventions	4
3	Technologies utilisées et conventions	5
3.1	Accord sur le système de gestion de bases de données	5
3.2	Versions des technologies	5
3.3	Conventions de nommage	5
3.4	Tailwind	5
3.5	Recommandation du format d'images	5

1 Discussion sur la communication, la "boîte noire" des deux projets

1.1 Service "GameService"

Dans un premier temps de l'échange a été établie la façon de communiquer entre les deux projets.

Pour cela, il a été choisi d'utiliser la notion de service. Il s'agira donc d'une classe, implantée par l'équipe s'occupant de la bibliothèque de jeux et utilisable par l'équipe s'occupant de la plateforme.

Ce service permettra notamment de créer une partie, d'en supprimer une, de rajouter des joueurs dans une partie ou d'en enlever.

Il sera possible de rajouter plus tard des méthodes supplémentaires si le besoin s'en fait ressentir.

1.2 Historique des coups

L'équipe s'occupant de la plateforme a notamment fait part de son besoin d'accéder aux historiques des coups de chaque joueur lors d'une partie. Il faut donc que l'équipe s'occupant des jeux enregistre en base de données les coups du joueur et rende accessible pour un joueur et une partie donnée la liste des coups effectués.

1.3 Informations sur les jeux, configuration du nombre de joueurs

Il a également été discuté de qui devait gérer les informations concernant les jeux, comme le nombre maximum et minimum de joueurs d'un jeu.

Une solution a été proposée de stocker l'information en dur du côté de la plateforme, mais il faudra probablement en rediscuter car il ne s'agit pas de la solution la plus propre. Il est envisageable d'avoir une BDD côté jeux pour stocker ces informations.

2 Discussion sur l'agencement du Git

2.1 Branche "développement"

Concernant l'agencement de l'espace de partage de code Git, les deux équipes se sont mises d'accord pour mettre en commun leur code directement depuis la branche de développement afin d'assurer le plus tôt possible le bon fonctionnement de l'application.

Ainsi chaque fonctionnalité développée par les groupes se basera sur la branche de développement pour créer sa propre branche. Quand la fonctionnalité sera finie et testée, un merge sera réalisé sur la branche de développement pour ajouter cette fonctionnalité.

2.2 Branche "pré-production"

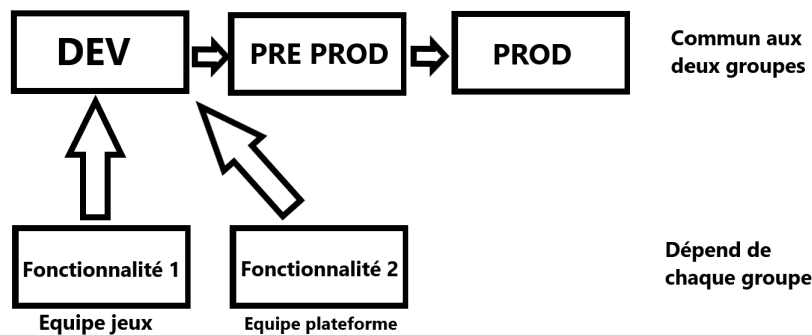
Si un ensemble de fonctionnalités liées est prêt et testé sur la branche de développement, et que le résultat est satisfaisant, alors cet ensemble peut être push sur la branche de pré-production.

Cette branche est d'une configuration similaire à la branche de production (qui constitue la mise en ligne du produit au client), il s'agit donc d'un test pour vérifier une dernière fois que l'ensemble est satisfaisant.

2.3 Branche "production"

A la fin d'un sprint lorsque les fonctionnalités sont présentes sur la branche de pré-production et qu'il est temps de livrer les attendus du sprint, ces fonctionnalités sont alors merge sur la branche de production.

On peut résumer ces explications en un schéma :



2.4 Commits et conventions

De plus, l'utilisation de gitflow a été proposée par l'équipe en charge de la plateforme pour permettre de réaliser plus efficacement le découpage des branches cité précédemment.

Enfin, il a été décidé qu'une norme précise de nommage des commits devait être réalisée afin que chacun puisse se retrouver dans la liste des commits et comprendre précisément et rapidement l'effet de chacun d'entre eux.

3 Technologies utilisées et conventions

3.1 Accord sur le système de gestion de bases de données

Tout d'abord concernant les technologies utilisées, les deux équipes du projet se sont mises d'accord sur l'utilisation de MariaDB pour la gestion des bases de données.

3.2 Versions des technologies

Il a également été décidé que les équipes devaient fournir la version des technologies qu'elles comptaient utiliser, notamment l'équipe des jeux doit communiquer les versions des technologies qu'elles a indiqué dans ses documents.

3.3 Conventions de nommage

Les équipes se sont également mises d'accord sur l'utilisation de SonarQube, et notamment Sonarlint qui est un plug-in à installer sur l'IDE de chaque développeur, permettant à tous les développeurs d'avoir les mêmes conventions de nommage. Il a de plus été évoqué l'utilisation des conventions BEM pour le nommage des classes.

3.4 Tailwind

Afin d'aider à la réalisation du CSS, le framework TailWind a été évoqué par le groupe travaillant sur la plateforme. Celui-ci sera utilisé par ce groupe, ce qui laisse la possibilité au groupe des jeux de l'utiliser ou non.

3.5 Recommandation du format d'images

Il a été recommandé par l'équipe de la plateforme d'utiliser les formats d'images webp ou avif dans le projet afin de permettre de meilleures performances pour l'application.