Réunion technique de revue de code du sprint 1

Agora - équipe jeux

16 février 2024

Participants

Nom	Prénom	Groupe
Stéphane	Buresi	Référent technique
Binginot	Etienne	1
Cauchois	Niels	1
Mazuy	Axelle	1
Montagne	Erwann	1
Thiberville	Malvina	1
Van Liedekerke	Florian	1

Ordres du jour

— Présentation de l'architecture du projet côté jeux
— Déploiement du projet
— Autres questions

Table des matières

1 Présentation de l'architecture du projet côté jeux

1.1 Modifications apportées à la bibliothèque

Des changements ont été effectués sur l'architecture de la bibliothèque de code depuis la revue de lancement du projet. En effet, la notion de GameController, un contrôleur réunissant les routes communes à tous les contrôleurs des différents jeux, a disparu car les routes étaient trop différentes et spécifiques entre elles.

M. Buresi a validé ce changement, il ne serait en effet pas propre de faire des structures conditionnelles dépendantes des jeux dans un contrôleur censé abstraire les jeux (de manière globale).

1.2 Améliorations possibles du code

Concernant les contrôleurs, M. Buresi nous a également conseillé de ne pas définir d'attributs en début de classe, mais plutôt de définir ceux-ci via les paramètres de constructeurs.

Il a également conseillé de ne pas renvoyer d'exceptions dans les cas ne représentant pas une erreur, mais de faire une gestion par code de retour.

D'un point de vue agencement du code métier, il a également été soulevé que certaines notions de code métier tel que le tri d'une liste ou certaines méthodes devraient davantage figurer dans un service et non dans un contrôleur.

Toutefois ces points ne représentent pas des erreurs en soi, et ne constituent donc que des améliorations pour le développement des jeux suivants.

1.3 Remarques sur l'architecture

M. Buresi a évoqué également la possibilité de découper en plusieurs services, si on voyait différentes notions métiers apparaître.

M. Buresi nous a également de nouveau conseillé de configurer PHPStan, ce qui est en cours de réalisation par l'équipe.

D'un point de vue global, M. Buresi nous a également indiqué que le code était bien typé et semblait propre. Il sera également à étudier la possibilité de lui donner un accès à notre GitLab pour qu'il puisse étudier plus finement notre code. Cela sera à voir avec M. Macadré, étant donné que l'invitation par e-mail est désactivée sur le GitLab.

2 Déploiement du projet

2.1 Serveur web à utiliser

Un point de discussion principal a été l'utilisation à prohiber du serveur web Symfony en déploiement. Bien qu'utile en local, ce serveur n'est pas conçu pour cela, ce qui le rend peu performant.

M. Buresi nous a également conseillé de ne pas l'utiliser en développement pour avoir le même environnement qu'en production en utilisant un Caddy server. En effet, les Caddy servers sont les serveurs les plus optimisés et leur utilisation est préconisée par la communauté Symfony.

2.2 Organisation des docker-compose

Il nous a également été conseillé de faire un docker compose par environnement avec un docker compose de base, commun à tous les environnements qui doit être écrit de manière variabilisée, et des fichiers plus spécifiques à chaque environnement.

Il ne faut également pas faire de volumes en environnement de production, parce qu'on ne va pas modifier les fichiers en production.

De même, M. Buresi nous a confirmé de bien conteneuriser pour déployer notre application.

2.3 Configuration de la CI/CD

M. Buresi nous a préconisé de réaliser un job de déploiement qui serait commun à tous les environnements de déploiement. Ce job devrait donc être variabilisé puis appelé dans les différentes étapes de la CI/CD avec des variables différentes.

3 Autres questions

3.1 Configuration de Mercure en HTTPS

L'équipe a présenté sa solution pour configurer l'HTTPS sur le projet à M. Buresi. Cette solution utilise un reverse-proxy (Traefik) qui configure l'HTTPS puis qui redirige sur les différents conteneurs en HTTP.

En effet la solution de base de Mercure, qui normalement peut configurer automatiquement l'HTTPS ne marche pas dans le cadre du déploiement sur les serveurs fournis par M. Macadré, ceux-ci n'étant pas accessibles depuis l'extérieur, et la création d'un certificat automatique ne marchant donc pas.

M. Buresi ne connaissant pas parfaitement cet aspect-là, nous a indiqué ne pas savoir s'il s'agissait de la meilleure solution mais de se reporter à la documentation. Mercure proposant l'utilisation de Traefik sur sa documentation, il semble s'agir d'une solution viable.

3.2 Les tests front-end

M. Buresi nous a indiqué la présence d'une documentation sur le plugin Symfony Panther, permettant de réaliser des tests sur la vue et de simuler des clics, etc.

3.3 Configuration de Mercure dans la CI/CD

L'équipe en charge des jeux s'est retrouvée face à un souci concernant la configuration de Mercure avec la CI/CD. En effet, Mercure n'étant pas lancé par la CI/CD, lors des tests d'application demandant de publier des notifications, une erreur se produit, Mercure étant inaccessible.

Une solution temporaire apportée serait de rediriger dans la CI/CD les requêtes Mercure vers le serveur de développement. Cependant, ce n'est pas la solution la plus propre, la CI/CD étant dépendante du serveur de développement.

Une autre solution serait de Mocker Mercure, mais M. Buresi ne sait pas si cela est possible. L'équipe se charge donc de se renseigner à ce sujet, et semble avoir trouvé une solution.