

RAGOT Cyrian (Equipe KL1)

Projet Long TOB : Rapport individuel n°1

DevOps et Product Owner

Lors de cette itération, je me suis investi dans le rôle de Product Owner et j'ai organisé et mis en place les outils de développement pour notre projet.

Dans un premier temps, j'ai fait un travail de recherche pour les librairies dont nous avons besoin. Après avoir comparé plusieurs librairies, JavaFX et JavaSound nous ont semblés être adaptées à notre projet.

Ensuite, j'ai créé le repository GitHub du projet et ajouté des règles dans les paramètres GitHub pour faciliter le travail en équipe. De plus, j'ai pris le temps de mettre en place une Github Action qui build notre projet avec Gradle et lance les tests unitaires lorsqu'une demande de merge est faite vers la branche principale. Cela évite que quelqu'un puisse merge alors que sa branche ne compile pas ou que les tests ne passent pas.

Notre projet est donc un projet Gradle, qui suis le format de répertoires de Gradle. J'ai alors mis en place les dépendances nécessaires et les paramètres de Gradle pour permettre à tout le monde de build/run/tester le projet de son côté en utilisant les commandes Gradle. En particulier, j'ai choisit d'utiliser Gradle 5.6 car cette version est compatible avec Java 11 et est installée sur les ordinateurs de l'école pour ceux qui souhaitent coder sur ces derniers. Mon but était de faire en sorte que le plus de personnes puisse facilement développer le projet en prenant en compte les différents IDE et environnements utilisés.

Tout au long de l'itération, j'ai pu aider mes camarades avec l'utilisation de git et de gradle.

Aussi, j'ai écrit un README qui indique quelques informations utiles aux développeurs.

Modèle du piano roll

Lors de cette itération, il m'a fallu lire beaucoup de documentation et de tutoriels sur l'API JavaSound pour développer des premières classes utilisant cette API.

En parallèle de Victor qui s'est occupé des tests unitaires, j'ai donc développé les classes Note.java et Pattern.java. Ces classes font partis du modèle du piano roll (selon le patron de conception MVC).

Ces classes permettent la manipulation et la sauvegarde des notes du piano roll qui pourront ensuite être jouées lors de l'implémentation avec le controlleur du piano roll.

Suite

- amélioration de la classe Pattern (optimisation et résolution de bugs)
- implémentation du lien entre Vue et Modèle à travers un Controller pour avoir un piano roll fonctionnel
- nouvelles features pour le piano roll
- autres features du projet