

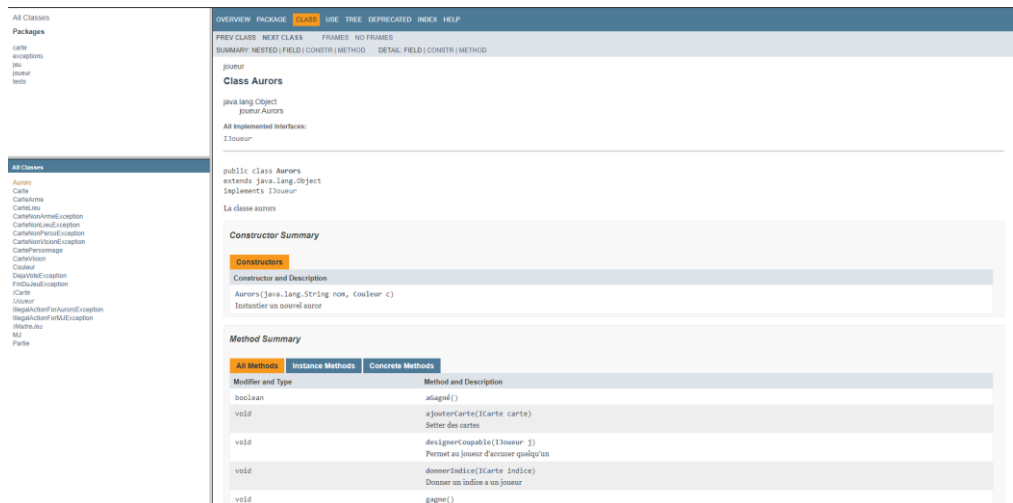
Compte Rendu 8

Ce compte rendu pour sur trois séances :

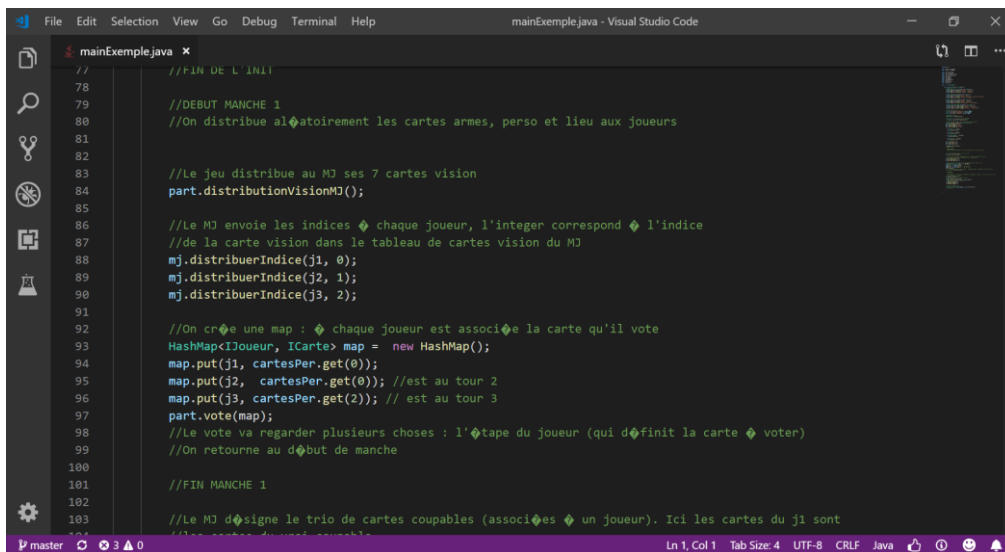
- Mercredi 20 Mars
- Jeudi 21 Mars
- Vendredi 22 Mars

Logique d'application

Etant donné que la logique d'application allait commencer à être utilisée par le corps du projet, nous avons développé des tests unitaires JUnit 5 et avons fait du débogage pour les passer. Une documentation précise, dotée d'exemples d'utilisation a été produite. Ainsi, le package de logique d'application a pu être intégrée dans le projet principal pour être utilisée.



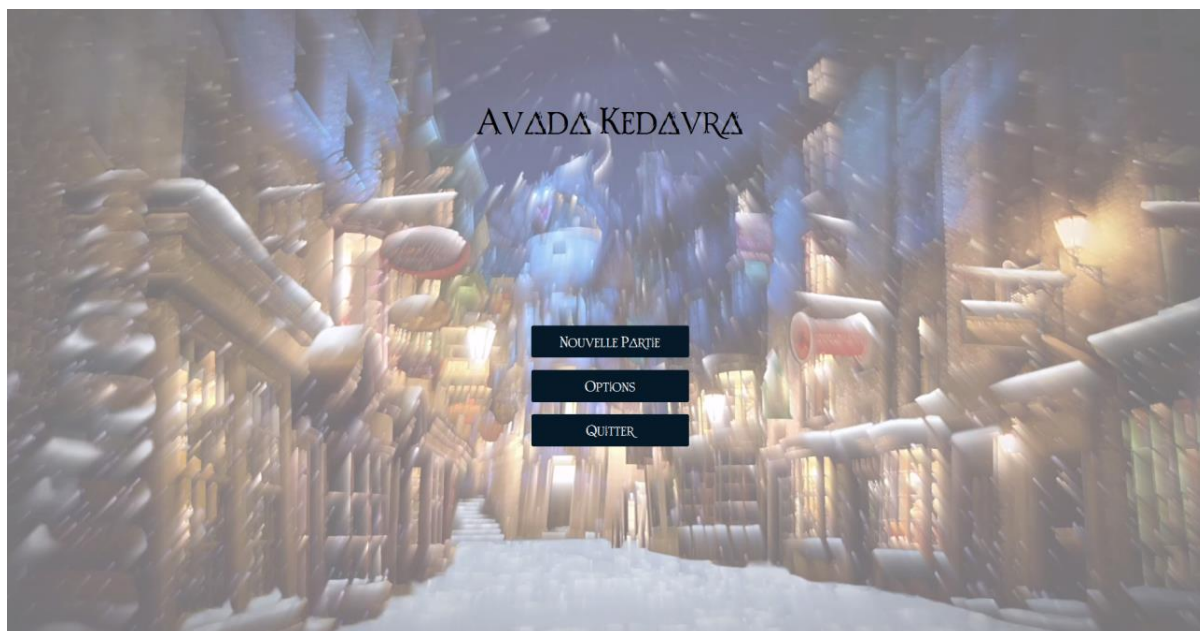
Documentation au format HTML



Exemple de déroulement d'un partie

Interface Graphique JavaFX

Nous avons travaillé sur l'aspect graphique de l'application pour la rendre plus agréable et moderne. Cela passe par la création d'un menu d'accueil, l'utilisation d'une police, de contrôles graphiques stylisés et d'un fond animé pour les vues d'accueil et de configuration. Cela a été réalisé entre autres grâce à l'utilisation et à l'intégration de fichiers FX CSS. Nous avons également doté le jeu de musiques d'ambiance.



De nouvelles vues de l'application ont également été développées, notamment la vue plateau, qui permet d'afficher les cartes qui ont été tirées aléatoirement par le jeu. Nous avons commencé à imaginer comment se ferait l'affichage des votes de chaque joueur.

Les assets graphiques n'étant pas encore terminés, nous avons créé des images de test pour déboguer. De même, dans une optique de débogage, nous avons développé un outil permettant d'ouvrir des vues indépendamment de l'application afin de pouvoir déboguer l'aspect graphique sans procéder à la mise en place des connexion réseau avec les clients.

Communication client-serveur

L'API de communication avec le client Android étant terminée, nous avons développé de nouveaux services réseau sur le serveur pour notamment :

- Envoi au MJ des cartes générées par la logique d'application pour chaque personnage
- Déroulement d'un tour de jeu (envoi de sa main, choix des indices par le MJ, envoi des indices au serveur, envoi des indices et des choix aux personnages et récupération des votes).

Ces services constituent l'intégration de la logique d'application.

Pour l'envoi par le réseau d'information complexe, (cartes, votes, main du MJ ...) nous avons choisi de faire des échanges de données au format JSON pour conserver la simplicité du flux en texte tout

en structurant nos données. Nous avons donc utilisé la bibliothèque JSON Simple côté serveur, et l'API JSON fournie par Android côté client.

Interface graphique Android

Les vues du MJ pour consulter les cartes des joueurs et choisir les indices sont fonctionnelles et nous avons commencé à développer les vues pour le vote des joueurs. Nous avons perdu un petit peu de temps à cause de débogueur intégré d'Android Studio qui faisait planter certains changements d'activité pour des raisons inconnues.

La vue MJ de consultation des cartes est un Slider créé à partir d'un contrôle graphique de ListView, nous avons donc développé un SliderAdapter et un layout de slider.

La vue MJ de choix des indices est également un Slider, mais dont les éléments sont cliquables de façon non exclusive.

Assets graphiques

Nous avons terminé de créer les cartes suspects, armes et lieux (parmi lesquelles les joueurs devront choisir) et avons commencé les cartes indices que le MJ devra distribuer aux joueurs pour l'aider dans son enquête. Pour cela nous avons cherché des images sur internet et les avons modifiées pour qu'elles aient toutes la même taille (format 16/9). Pour trouver les images nous avons utilisé des sites d'art comme DeviantArt ou Pinterest.