Compte rendu - 3ème séance PJS4

Au cours de la troisième séance de PJS4, nous nous sommes divisés par groupes afin de travailler sur les principaux points de notre application et développer le tout de manière simultanée. Notre objectif était de terminer le travail commencé la semaine précédente afin de pouvoir progresser le plus rapidement possible notre application et de préparer un premier jet afin de définir et trouver les différentes améliorations et modifications nécessaires. Selon le précédemment défini, nous nous sommes donc répartis de la manière suivante :

Logique Application & Test Unitaires	Interface JavaFX	Communication Client-Serveur	Interface Android	Assets Graphiques
Alexis	Thomas	Arsène	Sixtine	Personne
Gabriel		Maud		

Tâches réalisées :

Logique applicative :

Pour la logique liée à la conception et au déroulement d'une partie de jeu, nous avons réussi à terminer une première version qui permet de simuler la première phase du jeu. Cette logique étant terminée, il est à présent nécessaire de voir comment celle-ci peut s'appliquer sur nos applications graphiques et mobiles et de modifier au cours des prochaines séances le code réalisé. De plus, nous nous sommes concentrés sur la logique fondamentale et le code n'est pas réalisé de la manière la plus optimale possible, ce qui comptait était surtout de permettre au groupe s'occupant de la connexion et de la partie réseau de pouvoir avancer sur leur partie du projet. De plus, la javadoc et des tests unitaires et la gestion des exceptions ont été réalisés afin de s'assurer de la bonne maintenabilité et du bon fonctionnement de notre application. Enfin, la partie thread-safety sera réalisée la semaine prochaine avec la suite de la modélisation d'une partie normale.

Communication client-serveur :

Nous avons créé une classe *Appli* sur le serveur et une interface graphique de base qui permet de connecter les clients et de voir une progression lorsqu'un client se connecte puis lorsque tous les clients sont connectés.
Nous avons ensuite testé ces fonctionnalités avec un client BTTP basique sur un PC, ce qui marchait parfaitement. Nous sommes donc passé au test sur Android. Nous avons donc développé une classe encapsulant notre connexion ainsi qu'une vue d'application permettant de se connecter à partir de l'IP de l'hôte du serveur. Nous avons rencontré plusieurs problèmes, comme l'autorisation du trafic réseau pour notre application dans le SDK Android, ainsi que l'impossibilité d'utiliser des services réseaux sur le thread de l'interface (comme sur JavaFX). Nous avons résolu ces problèmes par l'autorisation des clés d'API dans le manifeste de l'application et le développement d'une *AsyncTask* qui permet d'utiliser un autre thread pour effectuer notre connexion au serveur.

Interface java FX :

Au cours de cette séance, nous avons pu nous familiariser un peu plus avec la bibliothèque JavaFX. Nous avons notamment imaginé le changement de “vue” de l’application, et avons développer une classe permettant d’ajouter, de supprimer et de changer de vue de manière rapide et efficace. Des tests unitaires ont aussi été réalisé. Pour la semaine prochaine, il faudrait à présent tester cette classe de manière graphique avec un cas pratique, afin de tester son bon fonctionnement.

Interface Android :

L’avancée de l’interface android a été difficile à cause d’une collision avec le module Firebase pour Android Studio (installé pour un autre projet). Après avoir cherché pendant un moment les erreurs, il s’est avéré que la base de données a altéré le bon fonctionnement de l’application Android et de l’intégrité du code. Une fois ce problème majeur réglé grâce à notre système de versionning, il a été nécessaire de remanier le code afin de respecter les normes et les usages de la programmation mobile.