GRAFF Mathieu
MARTIN-VILLEPOU Louis

Rapport SAÉ S1.01

Étape 1 :
Le premier réflexe qu'on a eu a été de créer un environnement de travail afin qu'on puisse travailler à 2 sur le projet. Pour cela, nous avons utilisé l'extension Live Share de Visual Studio Code afin de pouvoir éditer le code en même temps. Nous avons ensuite partagé le travail afin d'optimiser notre temps de travail. Pendant que l'un faisait le code principal, le deuxième faisait les tests sur ce code ou écrivait la javadoc.
La première étape était assez rapide et il n'y a pas eu de problème étant donné que la grosse partie a été faite pendant le TP11.

Étape 2 :
La deuxième étape était un peu plus longue, mais a été finie en environ 2 heures sans grosse difficulté.

Étape 3 :
Pour cette étape, nous avons eu du mal à comprendre l'énoncé et somme donc resté bloqués. Finalement, nous avons compris qu'il fallait créer un paquet vide et que la première carte doit être la plus petite si la pile est croissante ou la plus grande si elle est décroissante.

En refaisant quelques tests, nous nous sommes rendu compte qu'à un moment où la méthode insererTri est appelé il y avait une erreur. La méthode getValeur était appelé sur une carte null dans insererTri. Il fallait donc savoir quand est-ce qu'une carte null s'était introduite dans le paquet. L'erreur venait de la méthode piocherHasard qu'on avait modifié pour diminuer le nombre de return, mais on vérifiait si le paquet était vide après enlevé une carte. Donc quand il restait une seule carte dans le paquet, la méthode retournait forcément null alors qu'elle devait retourner la dernière carte.

Étape 4 :
Pour cette étape, nous avons rencontré un problème si il n’y avait plus de carte posable dans le premier paquet nous ne pouvions plus jouer et le jeu était déclaré comme perdu . Finalement ce problème venait de la méthode etreFini. Le parcours des cartes de la main sortait du tableau.

Conclusion :
Ce projet nous a permis d’approfondir nos connaissances des tableaux et de ne plus avoir aucun problème pour utiliser ces derniers. Nous avons aussi appris à mieux programmer en groupe en travaillant chacun de notre côté sur une tâche spécifique ou en regardant l'autre pour corriger ses erreurs.