

# Compte rendu

Guillaume PAQUIN et Rayane STASZEWSKI (groupe C)

## Progression actuelle

Cette première version s'arrête à la fin de l'étape 4 et contient un jeu fonctionnel

## Organisation du travail

Afin de mener à bien cette SAÉ, nous avons décidé de collaborer sur le code à l'aide de différents outils. Pour partager le code entre nous et effectuer des retours au cas où les fonctionnalités ajoutées casseraient notre programme, nous avons opté pour l'utilisation de [GitHub](#). En cours, nous avons cependant utilisé l'extension Visual Studio LiveShare, permettant de travailler sur le même code en même temps, de la même manière que fonctionne Google Docs. Ces outils nous ont permis de travailler en parallèle sur des questions différentes et d'optimiser le temps. Nous avons ainsi pu finaliser cette version 1 en un peu moins de quatre heures.

## Difficultés rencontrées

Pour coder l'algorithme des méthodes demandant plus de réflexion, comme la méthode [verifierCarteApres](#), nous avons utilisé la technique de la feuille et du stylo. Cela nous a permis de poser les différentes situations possibles pour l'utilisation de cette méthode et de décider comment l'algorithme devait réagir à chacune de ces situations. Une autre difficulté rencontrée fut sur la compréhension de certains éléments du sujet, comme par exemple si on devait ajouter ou non des cartes en début de partie à la frise, ou comment on pouvait placer une carte avant la première carte de la frise.

## Ajouts mineurs

Nous avons remarqué, durant la réalisation de notre projet, qu'il était impossible, avec les méthodes décrites dans le sujet, d'ajouter une carte avant la carte 0 placée dans la frise. C'est pourquoi nous avons décidé d'adapter la méthode afin que l'ajout en position 0 soit possible grâce à la position '-1'. De plus, pendant ce temps, mon camarade a ajouté de la couleur dans la console afin d'améliorer la lisibilité et de rendre le jeu plus agréable dans cet

environnement. Nous avons ajouté un deuxième paramètre lors de l'appel de `ProgJeu` afin de personnaliser le nombre de cartes donné à chaque joueur au début du jeu. Enfin, nous avons ajouté des délais entre chaque `println` afin de fluidifier l'expérience du joueur.

## La classe Jeu

Notre classe `Jeu` possède un constructeur prenant en paramètre le chemin d'accès au fichier `timeline.txt` et le nombre de cartes à donner à chaque joueur.

Le constructeur initialise les variables servant au jeu, ajoute au joueur ses cartes de façon aléatoire et place une carte aléatoire dans la frise.

La méthode `commencerJeu` permet de lancer la partie. Tant qu'il reste des cartes dans la pioche et dans les mains du joueur, on exécute des `tours` de jeu.

Un `tour` est décomposé en plusieurs instructions :

- L'affichage de la frise et de la main actuel
- La saisie d'une carte à prendre de sa main
- La saisie de l'emplacement où on veut mettre la carte
- Des affichages récapitulatif de la carte prise et de l'endroit où on veut la mettre
- Un affichage disant si la carte était bien placé ou non

Chacune de ces instructions ont été regroupées dans des méthodes qui leur sont propres.