

L'objectif de ce projet est de réaliser une application de flashcards qui sont largement utilisées comme un exercice d'apprentissage d'un sujet en reposant sur l'apprentissage actif active recall testing et la mémorisation par répétition espacée (spaced repetition). Il existe bon nombre d'applications de ce genre 1. Si vous n'êtes pas familier avec ce genre d'applications nous vous invitons vivement à étudier le fonctionnement des applications existantes et notamment la page de présentation de l'application **(Anki) 2** <https://docs.ankiweb.net/background.html#active-recall-testing> : Installez et essayez cette application.

Le mode “création” de l'application devra permettre :

- de créer une carte, d'éditer son contenu (face question, face réponse), de supprimer une carte ;
 - page avec affichage de toutes les cartes (mode global du carnet de voyage) sous forme de gridPane (on y affiche les réponses)
- de créer une pile (en précisant a minima un nom et une description), de modifier une pile, de supprimer une pile ;
 - Form de création de pile où sont affichées les cartes et où on peut cliquer dessus pour les ajouter à notre pile.
- d'exporter et d'importer une pile de cartes (dans votre propre format de stockage).
 - JSON ? (on ne pourra pas stocker les path d'image et de son parce que si la personne récupère juste le JSON, elle n'aura pas ces fichiers)
 - on fait un FileChoser pour importer et un FileChoser pour choisir le dossier où il exporte

Le mode “apprentissage” de l'application devra permettre :

- de travailler l'apprentissage d'une pile de cartes sélectionnée au préalable. L'application devra présenter la question, laisser un temps de réflexion, montrer la réponse, permettre à l'utilisateur de s'auto-évaluer sur la carte présentée ;
 - Page globale des piles dans la section “apprentissage”, l'utilisateur clique dessus, ça l'emmène sur la suite de questions de la pile (possibilité de faire affichage de réponses et il doit deviner la question, d'afficher le nombre de lettre de la réponse, la première lettre, en mode plusieurs niveaux de difficultés avec design pattern strategy)
- de configurer l'apprentissage en définissant différents paramètres influençant la répétition des cartes (l'algorithme de répétition, les fréquences de répétition, l'ordre de présentation) ;
 - Choisir entre ordre de la pile ou ordre aléatoire
 - Dans le cas d'un ordre aléatoire, choisir le nombre (une fourchette) du nombre de répétitions de chaque carte
- de consulter les statistiques d'apprentissage d'une pile ou de l'ensemble des piles. Ces statistiques pourront être présentées sous forme de graphiques.
 - chaque pile a une moyenne (un score, compris entre 0 et 1 avec 1 si le mec à 100% de bonne réponses et 0 si il a 0%)

Il est possible d'envisager un certain nombre d'extensions aux fonctionnalités basiques présentées ci-dessous :

- partager en ligne des piles de cartes (dans un répertoire GoogleDrive, un dépôt DropBox, un dépôt Git, etc.) ;
 - on fait pas
- intégrer des données multimédia à vos cartes (images, sons, vidéos, etc.) ;
 - embêtant pour les JSON (sous entend qu'on donne les fichiers vidéos / sons en même temps qu'on partage le JSON, embêtant)
- ajouter différents formats de rendu du contenu de vos cartes (rendu de formules mathématiques, de diagramme plantuml, etc.) ;
 - pourquoi pas, pour les maths ça se trouve avec une librairie quelconque je pense
- importer les piles de cartes provenant d'une autre application (par exemple le format .apkg d'Anki) ;
 - envisageable si on se penche sur le format des cartes de anki
- proposer une meilleure organisation des cartes et des piles de cartes (en partageant des cartes entre différentes piles, en ajoutant des étiquettes (tags) sur les cartes et les piles, en organisant les piles par leçons) ;
- modifier le style de présentation de cartes ;
 - éventuellement une couleur par catégorie si on crée nos propres catégories (via l'énumération)
- proposer un mode où l'apprenant doit saisir la réponse.
 - c'est pas dur mais pas pertinent car sensible aux fautes d'orthographe notamment

Trame :

- Situation initial : Donald Duck apprend (il est pas très bon)
- Il prend connaissance de l'application FlashDuck
- Et boom on lui montre un tuto (notre vidéo) qui explique les fonctionnalités de notre application
- PREMIÈRE PHASE : Mode création
 - Liste de piles déjà créées
 - Possibilité de créer une pile avec nom et catégorie
 - Catégorie système d'autocomplétion
 - Image par défaut pour les catégories personnalisées
 - Suppression / modification des piles
 - Ajout de carte
 - Question
 - Réponse
 - Modification / suppression de carte
- DEUXIÈME PHASE : Mode apprentissage
 - Choix de la pile d'apprentissage (impossibilité de choisir une pile vide)

- Lancer une partie
 - Mode affichage de question / réponse
 - utilité pédagogique (on apprend mieux comme ça)
 - Choisir de mélanger ou non le deck
- Durant la partie
 - Compteur qui affiche la réponse après 10 secondes automatiquement
 - bouton pour afficher directement la réponse
 - Possibilité d'indiquer si on avait ou pas la réponse
 - Si on a la réponse, elle est considérée comme validée et elle ne sera pas reposée
 - Sinon elle sera reposée avant la fin de la session
- A la fin de la partie
 - affichage du score de la partie
 - Possibilité d'aller à la page d'accueil
 - possibilité d'aller aux statistiques
 - Possibilité de rejouer la pile
- Statistiques
 - Pour chaque pile
 - graphique avec les différents scores
 - possibilité de supprimer les statistiques existantes
 - statistiques globales
 - affichage moyenne de chaque pile
- TROISIEME PHASE : les tests
 - montrer l'exécutions de tous les tests