

Conception et GdP, Coding Week

Encadrant : J. GUICHON

TELECOM NANCY
Projet de Conception et Développement
Deuxième année (S7)

2023, Semaine 1
Du 3 au 7 janvier

Groupe 28 :

Valentin CHANEL
Nicolas FRACHE
Théo GOUREAU
Cyrielle LACRAMPE-DITER

Table des matières

1	Conception	2
1.1	Classes	2
1.2	Controller et modèle	2
1.3	Controller view	2
1.4	Utils	2
2	Notre application	3
2.1	Fonctionnalités	3
2.2	Bugs connus	5
2.3	Améliorations possibles	5
3	Gestion de projet	6
4	Annexe	7
4.1	Diagramme de classes	7
4.2	Backlog	9

1 Conception

1.1 Classes

Le diagramme de classe simplifié est en Annexe (Figure 1). On y retrouve les trois parties du modèle MVC distinguées par des couleurs différentes.

1.2 Controller et modèle

Nous avons choisi de nous organiser autour du singleton Data qui sera notre lien entre la vue le modèle. Cette classe effectuera toutes les fonctions demandés sur les piles et s'occupera des opérations annexes pour préserver la cohérence et l'intégrité du modèle. De plus, c'est elle qui notifiera les observateurs.

Les modèles ont été divisées en trois catégories : piles, settings et statistiques ; ce qui a permis de rester organiser.

Une caractéristiques intéressantes dans notre implémentations des cartes est l'utilisation d'une classe QuestionRéponse assez général avec un type correspondant à un ENUM.

1.3 Controller view

Dans la partie controller, on retrouve classes gérant l'interaction entre le front FXML et la back avec la Classe Data faisant le pont entre eux.

A noté que plusieurs controller de vue utilise des classes de components pour les Cells des listView / tableView.

Nous avons également mis en place un pattern Factory pour l'aspect visuel des cartes. L'utilisation d'une classe général QuestionRéponse fait ici tout son sens : qu'elle que soit la données stockée, la factory va renvoyer un Pane contenant le média souhaité et il n'y a pas de différence de cas à faire pour le reste.

1.4 Utils

Enfin nous avons quelques classes nous permet de faciliter et factoriser des opérations complexes, notamment le FileHandler qui s'occupe de toutes les fonctions d'import / d'export des images et des piles sous format Zip. La classe calculs a

elle la responsabilité de toutes les opérations probabilistes nécessaire aux stratégies.

2 Notre application

2.1 Fonctionnalités

FlashBuddy aide ses utilisateurs à mémoriser des informations sur un sujet en lui posant des questions sur celui-ci.

L'application fonctionne avec un système de cartes : au recto se trouve la question, et lorsque l'utilisateur pense avoir la réponse, FlashBuddy retourne la carte pour dévoiler la réponse. L'utilisateur peut alors juger de s'il était proche ou non d'avoir la bonne réponse et ensuite passer à la prochaine carte. Les cartes peuvent être divisées en piles de carte correspondant à différents thèmes, pour pouvoir s'entraîner sur un sujet à la fois. Autour de ce principe, FlashBuddy propose différentes fonctionnalités.

Avant d'essayer de répondre aux questions, il faut indiquer les questions et leurs réponses à FlashBuddy. Voici les fonctionnalités permettant ceci :

- **Créer ses propres piles de cartes** - Notre application dispose d'un onglet édition, où il est possible de créer ou supprimer une pile, changer son nom, lui ajouter des "tags" pour indiquer le thème sur laquelle elle porte, et changer le nom de son auteur.
- **Nom d'auteur par défaut** - Dans les paramètres, il est possible d'indiquer un nom d'autre par défaut pour que toutes les piles créées par un même utilisateur indique son nom comme son auteur automatiquement.
- **Créer ses propres cartes** - Une fois notre pile créée, il est possible de l'éditer. On peut ainsi lui donner une description et lui ajouter ou enlever des cartes. Chaque carte dispose d'un champs contenant la question et d'un autre contenant la réponse.
- **Les questions et réponses sous différents formats** - Il est possible de présenter les questions et réponses sous forme écrite ou sous la forme d'une image. L'option image indique un bouton qui ouvrira l'explorateur de fichiers pour pouvoir choisir son image.
- **Import et export** - Afin de partager ses piles avec ses connaissances, il est possible d'exporter les piles créées par un utilisateur au format .zip ainsi que d'importer les piles créées par un autre utilisateur.

FlashBuddy est maintenant prêt à jouer, L'onglet apprentissage comprend plu-

sieurs options pour que les utilisateurs puissent apprendre dans les meilleures conditions :

- **Choix de la pile de cartes** - L'onglet apprentissage présente tout d'abord la liste des piles disponibles afin que l'utilisateur puisse choisir sur laquelle il veut s'entraîner. Il suffit de cliquer sur le bouton "Lancer l'entraînement" de la pile correspondante. Une fois l'entraînement lancé, il est possible de le quitter en cliquant sur "Terminer l'entraînement".
- **Choix de l'algorithme déterminant la prochaine carte** - Lors de l'apprentissage, FlashBuddy choisit une carte aléatoirement et en affiche la question. Il est possible de paramétrer l'application pour que le choix de la carte soit complètement aléatoire ou qu'il dépende de si la carte a été trouvée difficile ou non la dernière fois qu'elle a été jouée, pour voir plus souvent les cartes avec lesquelles l'utilisateur a du mal.
- **Répondre en temps limité** - L'utilisateur peut choisir d'afficher la réponse d'une carte lorsqu'il le souhaite avec le bouton "Afficher la réponse", mais aussi de faire en sorte que la réponse s'affiche automatiquement après une certaine durée : 5, 15 ou 30 secondes.
- **Noter la carte** - Après avoir vu la réponse, l'utilisateur peut juger de s'il a trouvé cette carte évidente, facile, moyenne, difficile ou impossible.

Après s'être entraîné sur ses cartes, l'utilisateur aura peut-être envie de savoir si sa performance a été satisfaisante. Cela est possible dans l'onglet statistiques :

- **Proportion de cartes pour chaque difficulté** - Il est possible de voir sous la forme d'un camembert la proportion de cartes que l'utilisateur a trouvé évidentes, faciles, moyennes, difficiles ou impossibles parmi toutes les cartes disponibles.
- **Proportion de cartes d'une pile pour chaque difficulté** - Le diagramme précédent est également disponible en prenant en compte une pile en particulier.
- **Nombre de cartes d'une pile jouées sur les 10 derniers jours** - Il est également possible de voir son activité sur les dix derniers jours sur une pile en particulier, c'est-à-dire le nombre de cartes jouées.

Pour personnaliser l'application selon ses goûts, ou selon l'heure de la journée, l'utilisateur peut choisir entre un thème jour et un thème nuit et soulager ses yeux lors de l'utilisation de FlashBuddy.

2.2 Bugs connus

Lors du test de notre application, nous avons remarqués quelques bugs que nous n'avons malheureusement pas eu le temps de résoudre avant le rendu de l'application.

- **Problème d'exportation de pile sous Windows** - Dans le cadre de l'utilisation de notre application sous Windows, si une pile contenant des images a été importée et que l'utilisateur essaye de l'enregistrer de nouveau (après une modification, par exemple), il y aura une erreur. Cela est dû à une portion de chemin codée en dur qui sera valide sous Linux, mais pas sous Windows. Nous savons d'où vient le problème et aurions pu le résoudre avec un peu plus de temps.
- **Texte lors du changement de forme d'une question/réponse** - Lorsqu'une question/réponse est sous la forme d'une image, si nous la changeons pour du texte, alors le texte est déjà rempli avec le texte d'une autre question au hasard.
- **Import des images** Sur certaines machines — notamment une sous Ubuntu 22.04 — l'import des images provoque des erreurs qui n'ont pas d'effets importants : il faut recharger la vue manuellement. L'import de l'image est bien réalisé mais une erreur a lieu avant l'appel des observateurs.
- **Affichage des caractères spéciaux** Sous certains OS, il est possible que certains caractères spéciaux apparaissent incorrectement.

2.3 Améliorations possibles

Bien que nous n'avons pas eu assez de temps pour offrir aux utilisateurs de FlashBuddy un large choix pour paramétrer l'application, nous avons fait en sorte qu'il soit facile d'en ajouter.

- **Choix de l'algorithme qui détermine la prochaine carte proposée lors d'un apprentissage** - Pour l'instant, il est possible de jouer avec une probabilité égale d'avoir chaque carte ou de pondérer ces probabilités avec la difficulté de chaque carte. Pour implémenter ce choix, nous avons utilisé le pattern Strategy ce qui rend l'ajout d'algorithmes supplémentaires assez simple. Nous pourrions par exemple forcer une carte à ne pas revenir avant une durée dépendant de si on l'a trouvée facile ou nous.
- **Affichage de différentes statistiques** - Nous gardons en mémoire tous les scores obtenus pour chaque carte non supprimée et la date et l'heure aux-

quelles la carte a été jouée, ce qui rend possible le calcul de statistiques différentes. Bien que nous n'ayons eu le temps de montrer que deux diagrammes différents, il existe beaucoup d'autres possibilités, comme indiquer la proportion de cartes faciles ou difficiles d'une pile sur la durée plutôt que de prendre en compte la dernière fois qu'une carte a été jouée seulement. Cela permettra à l'utilisateur de se rendre compte de sa progression.

- **Enregistrement des statistiques** - Nous avons malheureusement manqué de temps pour implémenter la sauvegarde des statistiques sous format JSON, les statistiques ne persistent donc pas entre deux utilisations de l'application, ce qui est dommage pour l'affichage de l'activité de l'utilisateur sur la durée.
- **Ajout de nouveaux médias** - Nous avons une architecture modulaire ce qui rendrait assez facile l'ajout d'audio ou de vidéo. Il suffirait d'ajouter un type et de créer les factory nécessaires. Il pourrait toutefois il y a avoir des difficultés lors de l'enregistrement sous format zip.

3 Gestion de projet

Nous avons utilisé un tableur pour vérifier que nous suivions bien la ligne directrice du projet et pour ne pas oublier les remarques : il est joint en annexe (cf. Section 4.2). Le projet étant court et puisque nous travaillions à côté les uns des autres, la communication était très directe et certaines tâches ne sont pas détaillées dans ce document.

Monsieur GUICHON venait faire le point avec nous trois fois par jour :

- Au début de la journée pour fixer les objectifs du jour
- En début d'après-midi pour faire le point sur l'avancement et pour nous aiguiller vers les objectifs principaux
- En fin de journée pour faire un bilan

Il était présent et de bons conseils tout au long de la semaine et nous a aidé à prioriser certains points du projet.

4 Annexe

4.1 Diagramme de classes

4.2 Backlog

Coding Week						
	Membres du groupe	Valentin CHANEL Nicolas FRACHE Théo GOUREAU Cyrielle LACRAMPE-DITER				
Nom de la tâche	Desc. de la tâche	Affectation de la tâche	Deadline	Update	Avancement	Commentaire
mardi 3 janvier 2023						
GDP01	Mise en place du Backlog	TG	11:00	10:58	100	
MVN01	Mise en place de Maven	NF	11:00	10:00	100	
DESIGN01	Dessin des différentes vues	CL	15:00	15:00	100	
CONC01	Diagramme de classe de base	VC, TG	16:30	16:16	100	
DEV01	Création des classes JAVA, déplacement entre les onglets	VC	16:21	15:11	100	
UI01	Base de l'interface avec header et onglets	NF		17:25	100	
DEV02	Classes vide du modèle	TG	18:00	17:50	80	Changement de conception, on reprendra l'implémentation demain.
CONC02	Modifier diagramme de classe	CL	18:30	18:30	100	
FH01	Création d'une classe util pour charger/sauvegarder les stack	VC	18:00	17:59	100	Non testé, le modèle n'a pas encore été construit
mercredi 4 janvier 2023						
DEV03	Implémentation de stats pour la nouvelle conception.	TG	10:40	10:40	100	Non testé.
DEV04	Implémentation simple des piles/cartes	TG	11:15	11:15	100	Non testé.
DEV05	Implémentation de Settings	TG	11:30	11:30	100	Non testé.
PEDIT01	Créer la page de modification des piles	VC	16:00	16:28	100	
DESIGN02	Page de settings	TG	16:30	16:33	100	Les boutons sont implémentés mais commentés car n'ont aucune action
DESIGN03	Ajout d'un thème clair	TG	17:30	17:10	100	
DESIGN04	Ajout du bouton d'ajout de pile	TG	18:00	18:01	100	
	Pattern factory pour ORView	CL			100	
	Vue apprentissage qui va chercher la cartes du modèle	CL			100	
PEDIT02	Ajouter les principales fonctionnalités de modification des piles	VC	18:30		100	
MAVEN02	Changement du plugin pour le fatjar (pour les libs des icônes)	NF		20:00	100	
jeudi 5 janvier 2023						
	Vue apprentissage qui met à jour les stats	CL	11:40	10:00	100	
	Choix des cartes avec proba en fonction des stats	TG	11:00	10:58	100	
	Ajout d'une description à la pile	TG	18:00	11:59	60	Ajouté dans le back et dans la view de l'édition
	Générer l'id de la pile	TG	11:40	11:40	100	
	Implémentation d'un timer pour le jeu	CL	18:00	14:00	100	
PEDIT03	Finaliser les édits de pile, y compris graphiquement	VC	15:00	12:00	68	
FH02	Amélioration du file handler, garde fou, optimisation, filtre	VC			100	
	Ajout de la liste de pile pour l'apprentissage et māj des données	NF	13:00		100	
	Choix des paramètre pour le timer	TG	14:30	14:30	100	
	Calcul des statistiques	TG	18:30	21:30	100	Non testé
	Génération des graphiques	TG	18:30	21:30	100	Non testé
FH03	Sauvegarde des piles sous forme de Zip	VC	18:00		100	
	stats-view et StatsController	CL	17:30	17:30	100	
FH04	Chargement des piles sous formes Zip	VC	23:00	01:30	100	
vendredi 6 janvier 2023						
TEST01	Si on importe deux piles de même id	VC			100	
OPT01	Date relative (il y a deux jours)				0	
	Pouvoir modifier l'auteur depuis le menu	NF			100	
FIX01	Réparer bug timer	CL		10:47	100	
TEST02	Ecrire une pile de test	CL		11:11	100	
	Ajouter nom d'auteur par défaut dans paramètres	CL		11:15	100	
FIX02	BarChart	TG	11:30	11:30	100	
CLEAN	Nettoyage de l'application suppression de fonctions inutiles	VC		12:00	100	
TEST03	Test des statistiques	CL	14:00	12:09	100	
TEST04	Test du modèle côté piles	CL		13:00	100	
	Test du modèle côté cartes	CL		13:18	100	
FIX03	Refresh manquant (stats)	TG	14:10	14:10	100	
DEV06	Suppression des piles/cartes aussi dans les stats	TG	14:30	14:30	100	
FIX04	Vérification de l'unicité des piles lors de l'import, création de nouvelle id sinon	VC		15:00	100	
DEV07	Nombre de boutons pour le score	TG	15:30	15:25	100	
FIX05	Stratégie de choix des cartes par probabilité	TG	15:51	15:50	100	Etablie comme stratégie par défaut
	Corriger taille des images dans apprentissage	CL			100	
	Refonte du design de la page Apprentissage	NF		17:00	100	
	Ajout d'un bouton pour supprimer les cartes lors de la modification d'une pile	NF		11:05	100	
FIX06	Correction pour l'import en place de ZIP (réécriture par dessus)	VC		17:30	100	
	Correction de problème de CSS sur l'ensemble de l'application	NF	18:00	17:30	100	
FIX07	Correction de problème de sauvegarde de pile importé de puis un ZIP	VC			100	
STATS01	Enregistrement des statistiques sous format JSON	VC			ECHEC	