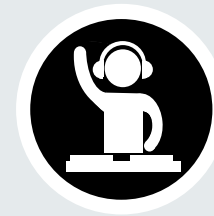


AUTOMIX SOFTWARE

COMPTE RENDU DE REUNION



DATE : 22/02/2018 N/RÉF :	OBJET : Réunion du Sprint 6
------------------------------	-----------------------------

DIFFUSION :	
PARTICIPANTS :	Baptiste Hemery Niels Hecquard, Anaëlle Troadec, Raphaël Herbert, Richard Nicolas, Cédric Masson
NON PARTICIPANTS :	Maxime Stevenot

COMPTE RENDU

1. Point n°1 : Discussion sur l'interface mix avancé (paramétrage du mix)	
1.1	Etre toujours dans un des 4 modes : par exemple dès qu'on modifie des paramètres par soi-même, il faudrait rajouter un 4 ^{ème} mode "advanced"
1.2	Bouton "apply" nécessaire ? Pas forcément, mais dans toutes les applis, c'est fait comme ça Rajouter logo sur la fenêtre
1.3	Parameter forms fait pour tous les paramètres (objectif à long terme)
2. Point n°2 : Discussion sur la modification de l'architecture	
2.1	Enregistrer truc dans fichier de config → utilisé par tout le monde → Problème : tout se trouve dans utils, toutes les préférences sont stockées dans un fichier xml
2.2	Solution : idée refaire un nouveau présentateur/modèle de paramètre (le présentateur est alors lié à un fichier xml, contenant toutes les préférences, lié à chaque ouverture de l'application) → manque une couche métier
2.3	Faire du présentateur un singleton, le rattacher à la fenêtre principale : depuis la fenêtre indépendante on accède au singleton → c'est comme ça qu'on doit accéder au présentateur Présentateur doit être indépendant de MainForm, remonte les infos à la Vue Modèle doit contenir des trucs sans intelligence (données pure et dures) MainForm réservé pour l'affichage → aspect visualisation Souci : il manque quelque chose qui gère les aspects métiers en background → rajouter une couche Enregistrement des paramètres sont pour l'instant dans le présentateur mais il devrait être dans une classe à part (dans un singleton), gère tout ce qui vient de la vue (là où on a trackcollection, param sur bpm, les dossiers temporaires, etc), charge au démarrage tout ça dans le modèle, partie vue l'alimente, la vue demande à la partie métier de lancer un tri, le métier demande à la vue de changer, etc Ici Présentateur fait les deux en même temps → il faut le scinder en 2 avec une partie métier qui s'occupe de garder trace de la track, rattaché à un ensemble de présentateurs, chacun des présentateur est rattaché à une form, partie métier forme les présentateurs qui décident de mettre à jour ou non les vues
3. Point n°3 : Discussion sur le paramétrage du tri	
3.1	Ajuster le coefficient, à normaliser dans simpletracksdistance (ligne 51) : ce qui a été normalisé c'est le paramètre bpmpriority et non pas la distance directement Ce qu'il vaut mieux faire : avoir un score sur chaque critère (bpm, clé) qu'on normalise, derrière multiplier par des coefficients qui ont un sens → travailler encore dessus