

Lecteur de diaporama – version du 28/03/2023

1 Description de l'application

1.1 Description fonctionnelle

Il s'agit d'un lecteur de diaporama, c'est-à-dire une application permettant d'afficher des images sous la forme d'un diaporama. Un diaporama est un regroupement d'images possédant un titre. Par exemple : « mes vacances 2021 ».

Le lecteur affiche les images du diaporama en cours. Chaque image est affichée accompagnée de son intitulé / titre, de la catégorie à laquelle elle appartient, et de son rang au sein du diaporama.

Deux modes de fonctionnement.

Le lecteur de diaporama dispose de 2 modes de fonctionnement :

- mode Manuel : L'image courante (et les informations qui l'accompagnent) restent affichées tant que l'utilisateur ne fait rien. Deux boutons (flèches « suivant » et « précédent ») permettent de passer manuellement à l'image suivante ou précédente. L'affichage 'boucle' : une fois sur la dernière image, le bouton « suivant » affiche la première image. De même, une fois sur la première image, le bouton « précédent » affiche la dernière image.
- mode Auto : Les images défilent automatiquement, à raison de 1 image toutes les 2 secondes. Le défilement automatique va uniquement vers l'avant, et démarre toujours à partir de la première image. L'affichage automatique 'boucle' : après la dernière image, l'application affiche à nouveau la première image.

Passage du mode Manuel au mode Auto et inversement.

A tout moment, il est possible de passer du mode Manuel au mode Auto, et inversement.

- Si l'on est en mode Manuel, le lancement du diaporama en mode Auto est possible, par clic sur un bouton « lancer diaporama ». Le défilement automatique démarre à la première image du diaporama (celle de rang 1 dans le diaporama).
- Si l'on est en mode Auto, la relance du diaporama en mode Auto est possible, par clic sur le bouton « lancer diaporama ». Le défilement automatique redémarre à la première image.
- Si l'on est en mode Auto, l'arrêt du diaporama automatique et le passage en mode Manuel est possible, par clic sur un bouton « arrêter diaporama », ou bien sur un des 2 boutons-flèches « suivant » / « précédent ». Le défilement automatique s'arrête, l'image qui était en cours d'affichage au moment du passage en mode Manuel reste affichée... jusqu'à ce que l'utilisateur décide d'interagir à nouveau avec l'application.
- Si l'on est en mode Manuel, le bouton « arrêter diaporama » est inactif.

Démarrage de l'application.

- L'application démarre toujours en mode Manuel. Le mode de fonctionnement en cours est toujours affiché sur la barre de statut.
- Lors du lancement de l'application, le lecteur est 'vide', à savoir aucun diaporama n'est chargé. Un menu « Paramètres -> Charger diaporama » permet de sélectionner le diaporama souhaité dans une base de données. L'application charge alors en mémoire le contenu du diaporama, c'est-à-dire son intitulé, ainsi que la liste des images qui lui sont associées.

Où sont les images ?

Les informations chargées dans le programme au sujet de chaque image sont les suivantes : intitulé / titre, catégorie, son rang dans le diaporama, et le chemin d'accès au fichier où se trouve le fichier contenant l'image.

Quant aux images, elles sont stockées localement, dans un dossier, sur la machine où s'exécute l'application.

Autres fonctionnalités.

Le menu « Paramètres -> Vitesse de défilement » permet de choisir le temps d’affichage de chaque image (en secondes) lorsque le diaporama est en mode Auto. Le temps d’affichage est propre à chaque Diaporama.

Le menu « Paramètres -> Enlever diaporama » permet de supprimer du lecteur le diaporama en cours de visualisation. Le lecteur revient alors à son état initial, vide.

Le menu « Fichier -> Quitter » permet de d’arrêter l’application, en ayant préalablement supprimé du lecteur le diaporama en cours de visualisation.

Le menu « Aide -> A propos de... » ouvre une boîte de Message indiquant les auteurs de l’application et la version de l’application.

La base de données.

La base de données contient les informations relatives aux diaporamas et aux images, sachant que :

- Un diaporama est décrit par son titre et la vitesse de défilement des images qui le composent
- Une image est décrite par un intitulé/titre, une catégorie, le chemin d’accès vers l’image stockées hors de la base de données.
- Une image peut faire partie de plusieurs diaporamas, à un rang différent.

La base de données vous sera fournie peuplée de quelques diaporamas. Les fichiers d’images sont également fournis.

1.2 Description de l’interface

Vous avez le choix quant à l’organisation de l’interface de votre lecteur.

Cette interface devra afficher tous les éléments demandés. Elle devra respecter les règles d’organisation des interfaces vues au S1, être ‘responsive’, rendre possible l’utilisation de l’application à l’aide de la souris et du clavier (tabulation – entrée), proposer des raccourcis clavier, etc..

2 Versions de l’application – développement incrémental

L’application fera l’objet de plusieurs versions. Les fonctionnalités seront ajoutées au fur et à mesure de votre avancement.

2.1 Caractéristiques de l’application dans sa version initiale

- La vitesse de défilement des images est rattachée au lecteur.
- Au lancement de l’application, un diaporama sera chargé par défaut.
- Pour les premières versions du programme, il n’y aura pas de base de données. Les données associées à un certain nombre de ‘tuples’ images seront chargées ‘en dur’ dans les objets du programme.
- Le lecteur fonctionnera tout d’abord en mode Manuel. Il intégrera le mode Automatique plus tard.

2.2 Table des versions

L’application sera développée de manière incrémentale. Chaque version intégrera une amélioration, soit fonctionnelle (= 1 nouvelle fonctionnalité), soit liée à l’organisation du code produit.

Chaque version est décrite dans le tableau suivant.

Version	Description fonctionnelle	Remarques éventuelles
v0	<p>Projet NON graphique (= projet console non Qt). Le lecteur s'exécute en mode console et ne propose que le mode Manuel. Le lecteur n'utilise qu'une classe Image, la notion de Diaporama n'existe pas en tant que classe.</p> <p>A faire : Codage du corps des classes C++ Image et Lecteur fournies. Le fichier main.cpp permet de tester le travail réalisé.</p>	
v1	<p>Projet Qt graphique. Seule l'interface graphique de l'application est complète et opérationnelle : tous les boutons et items de menus sont présents mais ils ne déclenchent pas les opérations métiers demandées : ils se contentent de communiquer (usage de qDebug()) qu'une action a été demandée par l'utilisateur.</p> <p>A faire Création avec le générateur d'interfaces de la classe fenêtre personnalisée LecteurVue dans un projet Qt graphique : fichiers lecteurvue.ui, lecteurvue.h, lecteurvue.cpp et main.cpp</p>	
v2	<p>Projet créé par duplication de la v2 Ajout des classes Image de la v0 dans le projet. Intégration dans la classe LecteurVue des fonctionnalités de la classe Lecteur de la v0</p> <ul style="list-style-type: none"> - Pas de base de données : création des images en dur, comme dans la v0 - Affichage des images dans l'interface graphique - Fonctionnement du lecteur en mode Manuel uniquement - Le menu Fichier >> Quitter permet d'arrêter l'application - Le menu Aide >> A propos de... ouvre une Boîte de Message donnant des informations sur : <ul style="list-style-type: none"> - la version de l'application - la date de création - les auteurs 	
v3	<p>Projet créé par duplication de la v2</p> <ul style="list-style-type: none"> - Mise en œuvre du mode Auto 	
v4	<p>Projet créé par duplication de la v3</p> <ul style="list-style-type: none"> - Implémentation de Fichier >> Paramétrer >> Vitesse ... : ouvre une Boîte de Dialogue permettant de modifier la vitesse de défilement des images - Au démarrage, aucun Diaporama n'est chargé - Implémentation de Fichier >> Paramétrer >> Charger diaporama ... : qui charge les images 'en dur' et les mets à disposition triées par ordre croissant de rang - Implémentation de Fichier >> Paramétrer >> Enlever diaporama ... : qui enlève le Diaporama en cours. Le lecteur est vide à nouveau. 	
v5	<p>Projet créé par duplication de la v4</p> <ul style="list-style-type: none"> - Chargement des informations relatives aux images à partir de la Base de Données 	
v6	<p>Projet créé par duplication de la v5</p> <ul style="list-style-type: none"> - Ajout d'une Classe Diaporama dans l'application - Nouvelle implémentation de Fichier >> Paramétrer >> Charger diaporama ... : qui charge les informations relatives au Diaporama et aux Images à partir de la Bases de Données 	
v7_BD	<p>Projet créé par duplication de la v6. Au choix :</p> <ul style="list-style-type: none"> - A partir de l'application, modifier dans la Bases de données la vitesse de défilement du diaporama, ou bien changer l'intitulé ou le chemin d'accès à l'image 	
V7_MVP	<p>Projet créé par duplication de la v1.</p> <ul style="list-style-type: none"> - Mise au point de la v2 en respectant le modèle MVP 	

6 Modalités de remise du travail

Chaque version fera l'objet d'un commit sur le dépôt Github. Le commit concernera toujours :

- 1 fichier .pdf, nommé **ac_LecteurDiaporama_Nometudiant1_Nometudiant2_Nometudiant3_TPx.pdf** contenant l'Analyse & Conception de votre application.
Exemple : ac_LecteurDiaporama_NodenotCarpentierBouix_TP6.pdf
Il sera complété de dépôt en dépôt
- Le contenu du répertoire contenant les sources de l'application, à l'exception du fichier .pro.user

Chaque commit devra obligatoirement contenir ces 2 livrables.

Chaque commit sera identifié en fonction de la version produite. Le détail sera donné dans le § 6.1 et 6.3.

6.1 Calendrier des remises

Elles correspondent aux semaines de saé avec présence des enseignants. Les équipes devront montrer qu'elles avancent d'une échéance à l'autre.

6.2 Le dossier d'analyse et de conception

La dernière version du dossier d'analyseEtConception de l'équipe sera déposée sur eLearn, dans la section associée à cette saé.

La page de garde contiendra : le titre de la saé, les auteurs, groupe de TP, année universitaire, année de formation, et **url sur Github** correspondant au lien vers votre dépôt public.

A l'intérieur, le dossier contiendra un sommaire, il y aura au moins autant de chapitres que de versions développées de l'application, et un dernier chapitre avec votre bilan :

- ce que vous avez appris,
- ce que vous avez aimé / pas aimé,
- ce qui a été difficile,
- le temps passé (sur conception/sur code),
- ce que vous auriez pu faire mieux (avec le recul),
- ce qui pourrait être amélioré dans la saé

➔ Par conséquent, prévoyez d'ores et déjà un tableau dans l'espace partagé du groupe avec le temps de travail passé par version par chaque membre. Le but n'est pas de vous contrôler, mais d'avoir une vision réelle du temps de développement nécessaire pour que nous puissions le comparer au temps prévu par la saé.

6.3 Versionnement – Git - Github

Ces informations seront détaillées ultérieurement.

7 Modalités d'évaluation

L'évaluation sera basée sur :

- le travail intermédiaire (analyseConception + code) montré par les étudiants aux enseignants durant les séances d'accompagnement prévues, et les réponses aux questions posées par l'enseignant sur ce travail
- le dossier final rendu par les équipes, et les versions produites
- la correspondance code <-> dossier
- la qualité du code produit
- le respect des règles d'ergonomie identifiées durant le projet
- si possible, le mini-oral prévu en fin de projet où chaque équipe présentera son travail aux enseignants.