

Lecteur de diaporama – version du 02/04/2024

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.

Exemples de fonctionnalités.

Un 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.

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

Un 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.

Un filtre de diapositives du diaporama courant, qui permet de n’afficher que les diapositives appartenant à une Catégorie donnée.

.... Vous pouvez ajouter les fonctionnalités de votre choix à ce noyau.

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.

Dans le diaporama, les images pourront être filtrées par catégorie.

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 (cf. S1 + documents fournis en R2.02), 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.

Il est recommandé d’assurer une gestion des versions à l’aide de l’outil **git**.

2.1 Caractéristiques de l’application dans sa version initiale

- Au lancement de l’application, un diaporama constitué d’une image unique est chargé par défaut.
- Pour les premières versions du programme, il n’y aura pas de base de données. Les données seront chargées ‘en dur’ dans les variables, puis 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 interne du code produit (exemples : passage de la v0 à la v1, v7).

Le tableau suivant présente, à **titre indicatif**, quelques exemples de versions. Les versions v1 et v2 sont obligatoires.

Version	Description fonctionnelle	Remarques éventuelles
v0 = version fournie	<p>Projet NON graphique (= projet console non Qt). Le lecteur s'exécute en mode console et ne propose que le mode Manuel. La version fournie est codée en mode procédural.</p> <p>A faire : Code : néant Dossier de conception : néant</p>	
v1	<p>Projet NON graphique Orienté Objets (= projet console non Qt). Version OO du programme précédent et conforme au diagramme de classes proposé. Le lecteur s'exécute en mode console et ne propose que le mode Manuel.</p> <p>A faire pour parvenir à cette version : Dossier de conception : <ul style="list-style-type: none"> - Réaliser le diagramme des classes UML sous-jacent à l'application : classes Lecteur, ... - Documenter les attributs et méthodes de chaque classe (signification / buts). Code : <ul style="list-style-type: none"> - Les classes figurant sur le diagramme UML sont implémentées. - Le fichier main.cpp permet de tester le travail réalisé : il crée les principaux objets, affiche leur contenu, saisit les demandes de l'utilisateur, et active les méthodes des objets en fonction des demandes de l'utilisateur. </p>	Dépôt : semaine civile 18
v2	<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 déclenchée par l'utilisateur.</p> <p>A faire pour parvenir à cette version : Dossier de conception : <ul style="list-style-type: none"> - Réaliser le diagramme états-transitions de l'application uniquement (comportement du lecteur), sous ses 2 formes (classique matricielle) - Documenter le lien entre éléments d'interface et fonctionnalités Code : <ul style="list-style-type: none"> - 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>	Dépôt : semaine civile 18
v2_MVP	<p>Projet créé par duplication de la v2, puis intégration des classes de la v1.</p> <ul style="list-style-type: none"> - Mise au point de la v2 en respectant le modèle MVP <p>A faire pour parvenir à cette version : Dossier de conception : <ul style="list-style-type: none"> - Réaliser le diagramme des classes UML sous-jacent à l'application : classes Lecteur, ... - Documenter les attributs et méthodes de chaque classe (signification / buts). Code : <ul style="list-style-type: none"> - Les classes figurant sur le diagramme UML sont implémentées. </p>	Dépôt : semaine civile 19
v3 / v3_MVP	<p>Projet Qt graphique, créé par duplication de la v2 (ou v2_MVP) Affichage des images dans l'interface visuelle. Intégration dans l'application de certaines fonctionnalités de la v1</p> <ul style="list-style-type: none"> - Pas de base de données : toujours création des images 'en dur' - 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 	Dépôt : semaine civile 19

	<p>A faire pour parvenir à cette version : Dossier de conception : - Mise à jour de la description de la version Code : - Mise en œuvre des fonctionnalités annoncées</p>	
v4 / v4_MVP	<p>Projet créé par duplication de la v3 (ou v3_MVP) - Mise en œuvre du mode Auto</p> <p>A faire pour parvenir à cette version : Dossier de conception : - Mise à jour de la description de la version Code : - Mise en œuvre de la fonctionnalité annoncée</p>	Dépôt : s20
v5 / v5_MVP	<p>Projet créé par duplication de la v4 (ou v4_MVP) - 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.</p> <p>A faire pour parvenir à cette version : Dossier de conception : - Mise à jour de la description de la version Code : - Mise en œuvre de la fonctionnalité annoncée</p>	Dépôt : s20
v6 / v6_MVP	<p>Projet créé par duplication de la v5 (ou v5_MVP) - Chargement des informations relatives aux images à partir de la Base de Données</p> <p>A faire pour parvenir à cette version : Dossier de conception : - Mise à jour de la description de la version Code : - Mise en œuvre de la fonctionnalité annoncée</p>	Dépôt : s21
v7 / v7_MVP	<p>Projet créé par duplication de la v6 (ou v6_MVP) - Chargement des diaporamas à partir de la base de données - Nouvelle implémentation de Fichier >> Paramétrer >> Charger diaporama ... : qui charge les informations relatives au Diaporama et aux Images à partir de la Bases de Données</p> <p>A faire pour parvenir à cette version : Dossier de conception : - Mise à jour de la description de la version Code : - Mise en œuvre de la fonctionnalité annoncée</p>	Dépôt : s22 Notifier dépôt aux enseignants
v8 / v8_MVP	<p>Projet créé par duplication de la v6 (ou v6_MVP) Au choix : 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</p> <p>A faire pour parvenir à cette version : Dossier de conception : - Mise à jour de la description de la version Code : - Mise en œuvre de la fonctionnalité annoncée</p>	Dépôt : s23

3 Ressources à disposition

Intitulé	Description – Localisation
LecteurDiaporamaAnalyseEtConception.doc	Un canevas de dossier d'analyse & conception. Vous devrez compléter ce document au fur et à mesure de l'avancement de votre projet. Il fera l'objet du même versionnement que l'application !
main.cpp	Version procédurale du noyau de l'application.
dossier images	Dossier contenant les images que nous avons utilisées pour réaliser l'application.

4 Equipes

Vous travaillerez par équipes de 3 ou 2 étudiants, à l'intérieur d'un même groupe de TP.

Groupe de TP	TP1	TP2	TP3	TP4	TP5
Effectif	12 ?	12 ?	11 ?	11 ?	14 ?
?? équipes					

Les équipes constituées s'inscriront sur eLearn selon la procédure habituelle (des saé 1.1 et 1.2) avant la seconde séance de saé.

Il leur sera demandé l'url vers le dépôt Github qui sera utilisé dans la saé. Le dépôt sera public et aura pour nom « LecteurDiaporama ».

5 Durée et planning des séances de la Saé

<p>La saé dure 8 semaines.</p> <p>Au total, elle prévoit 4 séances de TD et 11 de TP.</p> <p>Attention, le planning est susceptible de varier un peu afin de dégager un peu plus de temps en dernière semaine pour la partie évaluation.</p>	Sem. Form.	Sem. Civile	du Lundi	au Samedi	Précédentes Cours	Td-Tp	Dagorret TD 1.5h	TP 1.5h	Nodenot TD 1.5h	TP 1.5h	Pas d'enseignant TD 1.5h	TP 1.5h	Bouix TD 1.5h	TP 1.5h	Carpentier TD 1.5h	TP 1.5h	Remarques planification	TD 1.5h	TP 1.5h
	18	4	22/01/2024	27/01/2024	N/A	N/A												0	0
	19	5	29/01/2024	03/02/2024	N/A	N/A												0	0
	20	6	05/02/2024	10/02/2024	N/A	N/A												0	0
	21	7	12/02/2024	17/02/2024	N/A	N/A												0	0
	22	8	19/02/2024	24/02/2024	N/A	N/A												0	0
	9	26/02/2024	02/03/2024	Vacances d'hiver															
	23	10	04/03/2024	09/03/2024	N/A	N/A												0	0
	24	11	11/03/2024	16/03/2024	N/A	N/A												0	0
	25	12	18/03/2024	23/03/2024	N/A	N/A												0	0
	26	13	25/03/2024	30/03/2024	N/A	N/A					5						en amph1 tous	0	1
	27	14	01/04/2024	06/04/2024	N/A	N/A	1		1		5			1			séances	1	1
	28	15	08/04/2024	13/04/2024	N/A	N/A		1		2		5		1		1	séances	0	2
	16	15/04/2024	20/04/2024	Vacances de printemps															
	17	22/04/2024	27/04/2024	Vacances de printemps															
	29	18	29/04/2024	04/05/2024	N/A	N/A	1		1		5		1					1	1
	30	19	06/05/2024	11/05/2024	N/A	N/A		1		1				2		1		0	1
	31	20	13/05/2024	18/05/2024	N/A	N/A	1		1		5				1			1	1
	32	21	20/05/2024	25/05/2024	N/A	N/A		1		2		5			1			0	2
	33	22	27/05/2024	01/06/2024	N/A	N/A					5							0	1
	34	23	03/06/2024	08/06/2024	N/A	N/A	1	3		1	3			1	2		2	1	2
	TDs : en début																		

6 Modalités de remise du travail

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

- 1 fichier **pvxx_LecteurDiaporama_Nometudiant1_Nometudiant2_NomEtudiant3_TPx.pdf (ou .txt)** de Présentation de la version vxx, décrivant rapidement les caractéristiques de cette version de l'application.
- 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 3 livrables.

Chaque commit sera identifié en fonction de la version produite.

6.1 Calendrier des remises

Les versions v1 et v2 ou v2_MVP sont à remettre obligatoirement au plus tard le lundi 29/04.

Elles feront l'objet d'une **présentation & évaluation** par des enseignants durant la semaine civile 18.

L'évaluation sera individuelle.

Le calendrier des remises est précisé dans la table des versions.

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

Sur demande, des informations pourront être fournies ultérieurement.

7 Modalités d'évaluation

L'évaluation sera basée sur :

- **La présentation-évaluation** des versions v1 et v2/v2_MVP
- 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
- Dans la mesure du possible, une dernière évaluation-présentation en fin de projet où chaque équipe et individu dans l'équipe présentera son travail aux enseignants.