

Saé 2.01 – Développement d’une application

Lecteur de diaporamas – Dossier d’Analyse et conception

1. Compléments de spécifications externes.

Aucun point flou n’a été à déclarer, le sujet est clair et compréhensible.

2. Scénarios

Description du scénario nominal et de un / deux scénarios alternatifs afin de mettre en évidence les interactions entre le système et l'utilisateur

Scénario nominal numéro 1:

Cas d'utilisation	Mode Manuel Diaporama	
Acteur primaire	Utilisateur	
Système	SYSTÈME ETUDIE	
Acteur secondaire	---	
Préconditions	l'application est lancé et le diaporama est chargé	
Opérations	Acteur	Système
1	l'utilisateur demande de voir l'image suivante	
2		le système affiche l'image suivante
3	l'utilisateur demande de voir l'image précédente	
4		le système affiche l'image précédente
Extension		
1.A	l'utilisateur demande de voir l'image suivante en étant à la dernière	
1.A.2		le système affiche la première image
3.A	l'utilisateur demande de voir l'image précédente en étant à la première	
3.A.2		le système affiche la dernière image

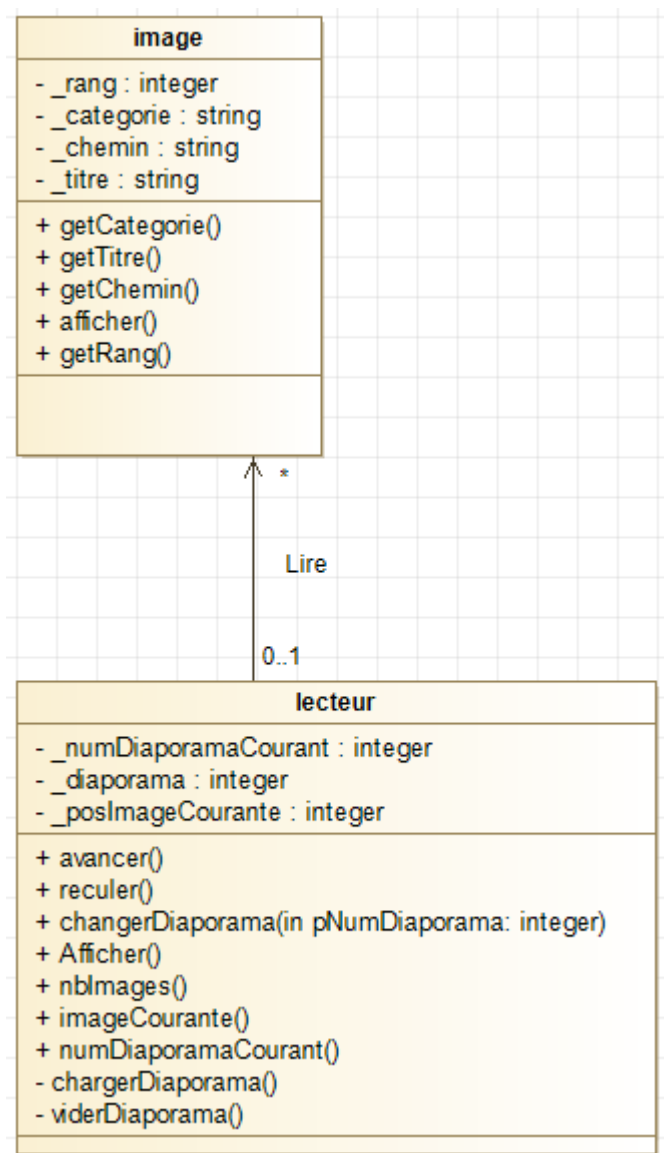
Scénario nominal numéro 2:

Cas d'utilisation	Mode automatique Diaporama	
Acteur primaire	Utilisateur	
Système	SYSTÈME ETUDIE	
Acteur secondaire	---	
Préconditions	l'application est lancé et le diaporama est chargé	
Opérations	Acteur	Système
1	l'utilisateur demande a lancer le diaporama alors qu'il est en mode manuel	
2		le système active le mode auto
3		le système affiche la premiere image et passe a la suivante toutes les 2 seconde
4	l'utilisateur click sur "arreter le diaporama"	
5		le système active le mode manuel
Extension		
1.A	l'utilisateur demande a lancer le diaporama alors qu'il est en mode auto	
1.A.2		le système affiche la premiere image et passe a la suivante toutes les 2 seconde
4.A	l'utilisateur click pour passer a l'image suivante	
4.A.2		le système affiche l'image suivante
4.A.3		le système passe en mode manuel
4.B	l'utilisateur click pour passer a l'image précédente	
4.B.2		le système affiche l'image précédente
4.B.3		le système passe en mode manuel

3. Diagramme de classe (UML)

(a) Le diagramme de classes UML se focalise sur les classes **métier**, cad celles décrivant les éléments structurants de l'application, indépendamment des éléments d'interface.

Voici le diagramme de classe (UML) de la version V0 de notre application.



(b) Dictionnaire des éléments pour chaque classe

Classe image			
Nom attribut	Signification	Type	Exemple
_rang	Rang de l'image au sein du diaporama auquel l'image est associée	unsigned int	1

_titre	Intitulé de l'image	String	
_categorie	Catégorie de l'image	String	personne
_chemin	Chemin complet vers le dossier où se trouve l'image	String	
Nom Sous programme	Signification	Type	
Image(unsigned int pRang=0, string pCategorie="", string pTitre="", string pChemin = "")	Constructeur par défaut de la classe image	Image	
getRang()	Renvoie le rang de l'image		
getCategorie()	Renvoie la catégorie de l'image	String	
getTitre()	Renvoie le titre de l'image	String	
getChemin()	Renvoie le chemin de l'image dans les dossiers	String	
afficher()	Affiche tous les champs de l'image	void	

Classe lecteur			
Nom attribut	Signification	Type	Exemple

<code>_numDiaporammaCourant</code>	Numéro du diaporama courant, par défaut 0	unsigned	1
<code>_diaporama</code>	Pointeurs vers les images du diaporama	Diaporama	1
<code>_posImageCourant</code>	Position, dans le diaporama, de l'image courante. Indéfini quand diaporama vide. Démarre à 0 quand diaporama non vide	unsigned int	
Nom Sous programme	Signification	Type	
<code>avancer()</code>	Procédure afin de passer à l'image suivante dans le diaporama	void	
<code>reculer()</code>	Procédure afin de passer à l'image précédente dans le diaporama	void	
<code>changerDiaporama()</code>	Procédure afin de choisir un diaporama et de le charger dans le programme	void	
<code>Afficher()</code>	Procédure afin d'afficher les informations sur lecteur-diaporama et image courante	void	
<code>nbImage()</code>	affiche la taille de <code>_diaporama</code>	unsigned short int	
<code>imageCourant()</code>	retourne le pointeur vers l'image courante	Image*	
<code>numDiaporamaCourant()</code>	retourne le nombre de diaporama que le programme peut charger	unsigned short int	

chargerDiaporama()	charge dans _diaporama les images du _numDiaporamaCourant	void	
viderDiaporama()	Vide _diaporama de tous ses objets image et les delete	void	

Tableau 2 : Dictionnaire des éléments - Classe xxx

(c) Dictionnaire des méthodes : vous pouvez fournir directement le fichier entête de chaque classe.

Exemple (classe lecteur de la version Console) :

```
#ifndef LECTEUR_H
#define LECTEUR_H
#include "image.h"
#include <vector>

typedef vector<Image*> Diaporama;    // Structure de données contenant les infos sur
les images

class Lecteur
{
public:
    Lecteur();

    void avancer();                // incrémente _posImageCourante, modulo nbImages()

    void reculer();                // décrémente _posImageCourante, modulo nbImages()

    void changerDiaporama(unsigned int pNumDiaporama); // permet de choisir un
diaporama, 0 si aucun diaporama souhaité

    void afficher();                // affiche les informations sur lecteur-diaporama et
image courante

    unsigned int nbImages();        // affiche la taille de _diaporama

    Image* imageCourante();        // retourne le pointeur vers l'image courante

    unsigned int numDiaporamaCourant();
```

```

private:

    unsigned _numDiaporamaCourant;    // numéro du diaporama courant, par défaut 0

    Diaporama _diaporama;            // pointeurs vers les images du diaporama

    unsigned int _posImageCourante;    /* position, dans le diaporama,

                                         de l'image courante.

                                         Indéfini quand diaporama vide.

                                         Démarre à 0 quand diaporama non vide */

private:

    void chargerDiaporama();          // charge dans _diaporama les images du
    _numDiaporamaCourant

    void viderDiaporama();            // vide _diaporama de tous ses objets image et les
    delete

};

#endif // LECTEUR_H

```

Figure 4 : Schéma de classes = Classe XXX

(d) Remarques concernant le schéma de classes

1. On ne s'intéresse qu'aux attributs et méthodes métier. Notamment, on ne met pas, pour l'instant, ce qui relève de l'affichage car ce sont d'autres objets du programme (widgets) qui se chargeront de l'affichage. Par contre, on n'oublie pas les méthodes getXXX(), qui permettront aux objets métier de communiquer leur valeur aux objets graphiques pour que ceux-ci s'affichent.
2. On n'a mis ni le constructeur ni le destructeur, pour alléger le schéma.
3. D'autres attributs et méthodes pourront venir ultérieurement compléter cette première vision ANALYTIQUE de l'application. Il s'agira des attributs et méthodes dits DE CONCEPTION nécessaires au développement de l'application.

Version v0 – Version console seule

4. Implémentation et tests

4.1 Implémentation

Liste et rôle des fichiers de cette version :

lecteur.h	Spécification de la classe Lecteur
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image
image.cpp	Corps de la classe Image
main.cpp	Tester les méthodes de la classe Lecteur

4.2 Test

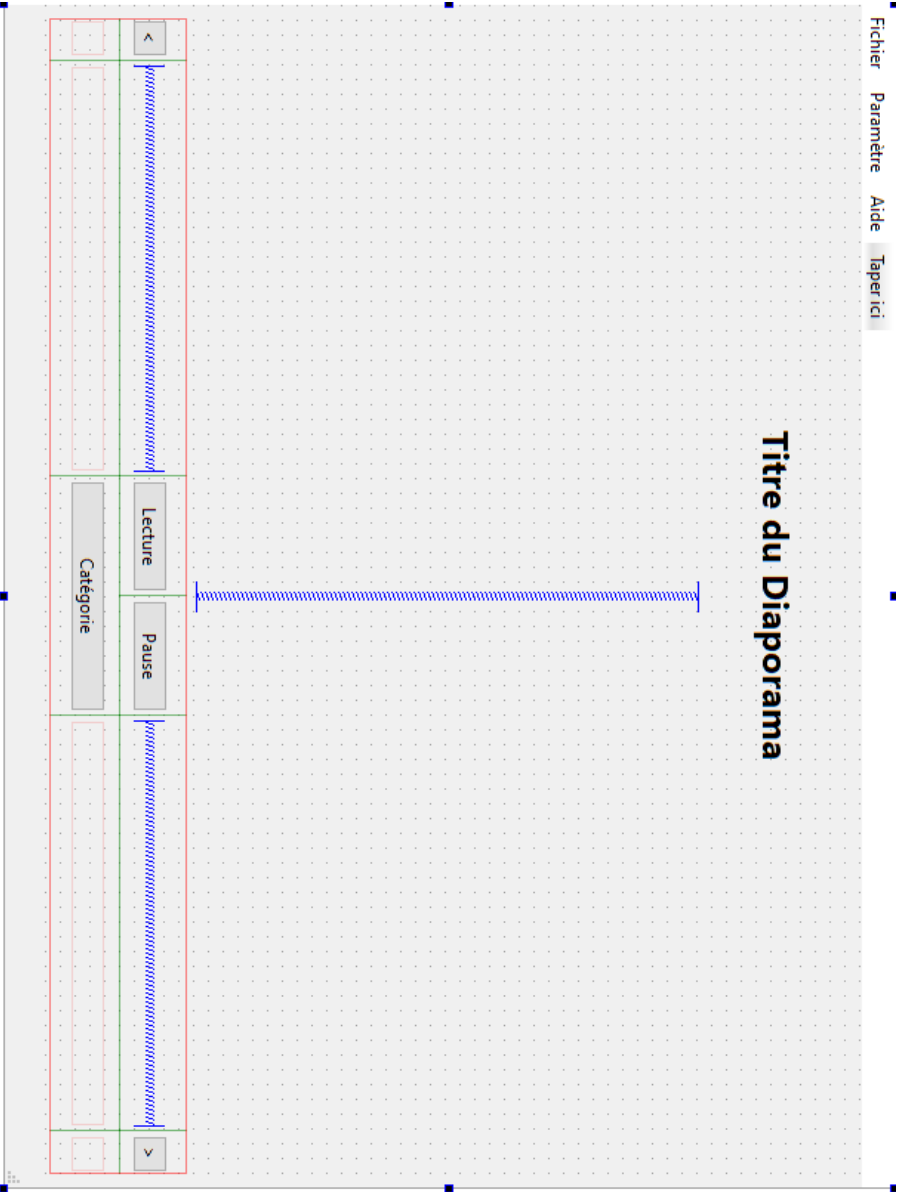
Description scénario	Résultats attendus		Réalisation test 1			Résultats test 2		
	trouvé	pos	résultats OBTENUS		Remarques : description du problème constaté lors de l'exécution - Idées de solution	résultats OBTENUS		Remarques
			trouvé	pos		trouvé	pos	
le système doit avancer 4 fois d'image	affichage de la console : "Test avancer() : 4 fois avancer() : avancer() : avancer() : " avancer() : "	---	affichage de la console : "Test avancer() : 4 fois avancer() : avancer() : avancer() : " avancer() : "	---	OK		---	
le système doit reculer 5 fois d'image	affichage de la console : "Test reculer() : 5 fois reculer() : reculer() : reculer() : " reculer() : "	---	Affichage de la console : " test reculer() : 5 fois"	---	non ok	affichage de la console : "Test reculer() : 5 fois reculer() : reculer() : reculer() : " reculer() : "	---	Probleme de boucle, une seule itération
le système enlève le diaporama de l'application une fois les tests faits	Affichage de la console : "Enlever le diaporama courant = Choisir diaporama 0"	---	Affichage de la console : "Enlever le diaporama courant = Choisir diaporama 0"	---	ok		---	
le système affiche le nom du diaporama	affichage du nom du diaporama au chargement du diaporama	---	affichage du nom du diaporama au chargement du diaporama	---	ok		---	
le système affiche le nom de l'image	affichage du nom de l'image au chargement de chaque image	---	affichage du nom de l'image au chargement de chaque image	---	ok		---	

Version v1 – projet Graphique seul

5. Éléments d'interface

A faire ici : description sommaire des éléments de l'interface, par exemple, avec une copie d'écran sur laquelle sont nommés les variables/objets graphiques et où les layouts sont positionnés et nommés.

Vérifier que tous les éléments graphiques qui seront manipulés par l'application ont des noms pertinents et bien formés.



Filter

Objet	Classe
▼ LecteurVue	QMainWindow
▼ centralwidget	QWidget
▼ glayoutBoutons	QGridLayout
hSpacer1	Spacer
hSpacer2	Spacer
pCategorie	QPushButton
pDroite	QPushButton
pGauche	QPushButton
pLecture	QPushButton
pPause	QPushButton
Titre	QLabel
vSpacer	Spacer
▼ menubar	QMenuBar
▼ menuAide	QMenu
actionA_propos_de	QAction
▼ menuFichier	QMenu
actionQuitter	QAction
▼ menuParam_tre	QMenu
actionCharger_diaporama	QAction
actionEnlever_diaporama	QAction
▼ menuVitesse_de_d_filament	QMenu
actionx0_5	QAction
actionx1	QAction
actionx2	QAction
statusbar	QStatusBar

6. Implémentation et tests

6.1 Implémentation

Liste et rôle des fichiers de cette version :

lecteurVue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas
lecteurVue.cpp	Corps de la classe LecteurVue
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner
main.cpp	Tester les méthodes de la classe Lecteur

Remarques sur l'implémentation

Sur cette version aucun bouton est lié à un signal, dû au fait que la classe lecteur ne n'est que la représentation graphique sur la fenêtre des boutons. Il n'y donc pas de SLOTS et de SIGNAL.

6.2 Test

A faire :

Décrire les tests prévus / réalisés pour montrer :

- Le comportement de l'interface non lié aux aspects fonctionnels du programme
- Le comportement de l'interface liée aux aspects fonctionnels du

V1 jeux de test:

recherchePremiereOccDecEntier

Réalisation test 1			
résultats OBTENUS		Remarques : description du problème constaté lors de l'exécution -	

	Le programme affiche la barre d'outils avec le bouton Aide contenant le bouton "A propos de"	La création et affichage du bouton "Aide" dans la barre d'outil contenant le bouton "A propos de"	---		La création et affichage du bouton "Aide" dans la barre d'outil contenant le bouton "A propos de"	---	ok
	Le programme affiche le titre du diaporama qui dans cette versions n'est pas chargé.	Affichage du titre du diaporama qui dans cette versions n'est pas chargé.	---		Affichage du titre du diaporama qui dans cette versions n'est pas chargé.	---	ok

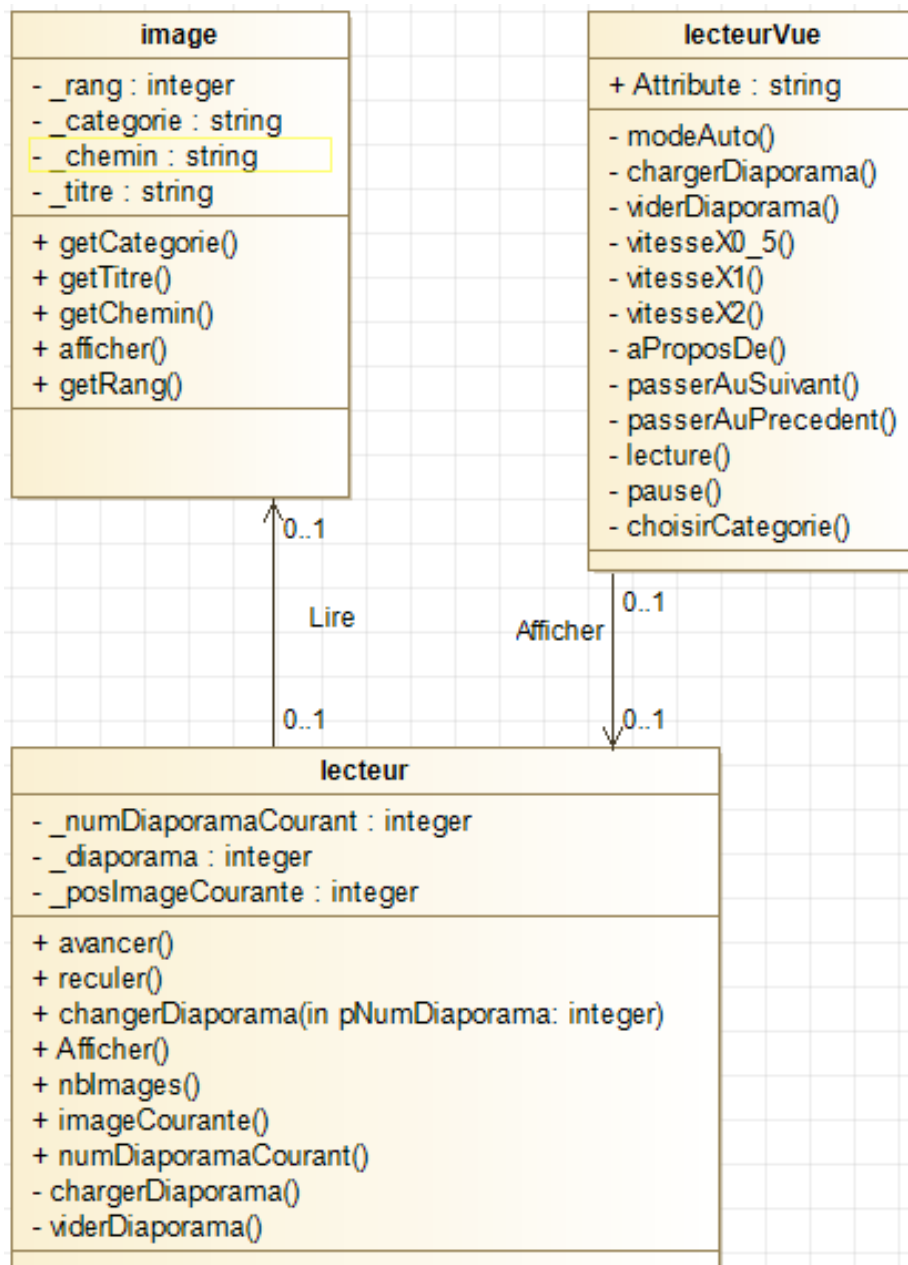
Remarque :

Il n'y pas de donnée rentrant c'est pourquoi nous ne les avons pas représentés, de même pour les deuxièmes résultats que nous avons décidé de ne pas représenter lorsqu'ils étaient absents.

Version v2 –

7. Diagramme de classes (UML)

A faire – s'il y a des changements - sinon indiquer que idem v0



8. Comportement de l'application

7.1 Diagramme états-transitions-actions du lecteur de diaporamas (v2)

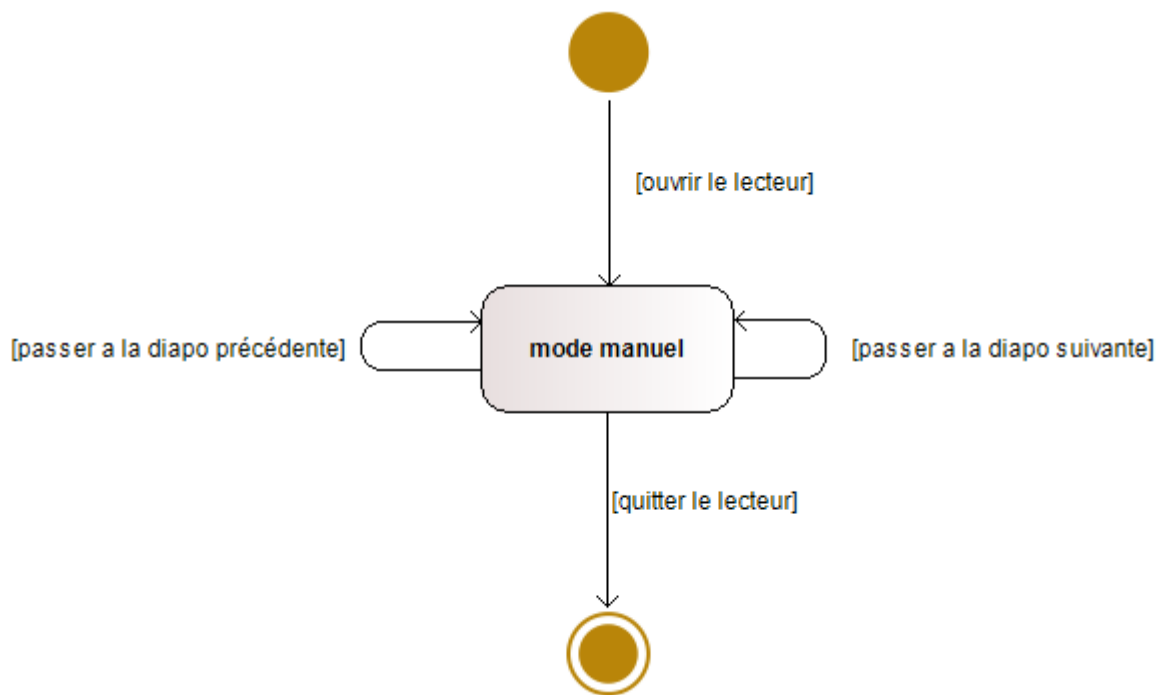


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v2

7.2 Dictionnaire des états, événements et Actions (v2)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
mode manuel	Le diaporama est en mode manuel (l'utilisateur passe les images manuellement)

Tableau 2 : États du lecteur de diaporamas – v2

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomEvénement</i>	<i>Signification</i>
---------------------	----------------------

ouvrir le lecteur	permet de charger le diaporama et donc d'initialiser automatiquement à son premier état qui est en mode manuelle
Quitter le diapo	Le programme quitte le diaporama

Tableau 3 : Evénements faisant changer le diaporama d'état – v2

Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>
<i>passer à la diapo suivant</i>	le diaporama passe à l'image suivante
<i>passer à la diapo précédant</i>	le diaporama passe à l'image précédente

Tableau 4 : Actions à réaliser lors des changements d'état – lecteur de diaporamas v2

7.3 Table T_EtatsEvenementsActions (v2)

Correspondance matricielle du diagramme états-transitions de l'application :

- en ligne : les **états** du lecteur de diaporamas (éventuel état de départ d'une transition)
- en colonne : les **événements** faisant changer le lecteur d'état (déclencheur d'une transition)
- dans chaque cellule : l'état d'arrivée de la transition + action/traitement à faire + éventuellement garde accompagnant la transition

<i>Élément graphique pregnant en charge cet événement à</i>			
---	--	--	--

<i>Événement à nomEtat</i>			
<i>passer à la diapo suivant</i>	l'utilisateur clique sur le bouton représenté par une flèche droite.		
<i>passer à la diapo précédant</i>	l'utilisateur clique sur le bouton représenté par une flèche gauche.		
<i>Quitter le diapo</i>	l'utilisateur clique sur Fichier puis sur quitter.		

Tableau 5 : Matrice d'états-transitions du lecteur de diaporamas – v2

L'intérêt de cette vue matricielle est qu'elle permet une préparation naturelle et aisée de l'étape suivante de programmation.

9. Implémentation et tests

8.1 Implémentation (v2)

Liste et rôle des fichiers de cette version :

lecteurVue .h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas. Permet d'afficher les attributs de la classe lecteur (diaporama)
------------------	--

lecteurVue. .cpp	Corps de la classe LecteurVue.
lecteurvue. ui	Fichier du dessin de l'interface réalisé par QtDesigner
lecteur.h	Spécification de la classe Lecteur. Déclare les procédures et les fonctions qui permettront de gérer l'affichage et des boutons.
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image. qui permet de gérer les informations d'une image pour les utiliser dans d'autres sous-programmes.
image.cpp	Corps de la classe Image
main.cpp	Permet d'exécuter le programme et d'afficher les fenêtres

Remarques sur l'implémentation

Tous les slots sont géré par la classe lecteurVue. Ce qui permet de réunir la gestion de tous les signaux des boutons.

8.2 Tests (v2)

V3 jeux de test:

```
recherchePremiereOccDecEntier
```

recherchePremiereOccDecEntier						Réalisation test 1			Résultats test 2		
	Description scénario	Résultats attendus		résultats OBTENUS		Remarques : description du problème constaté lors de l'exécution - Idées de solution		résultats OBTENUS			
		trouvé	pos	trouvé	pos			trouvé	pos		Remarques
	l'utilisateur clic sur "<"	Le programme passe a l'image précédente et passe le diaporama en mode manuel	---		Le programme passe a l'image précédente et passe le diaporama en mode manuel	---	ok				
	l'utilisateur clic sur ">"	Le programme passe a l'image suivante et passe le diaporama en mode manuel	---		Le programme passe a l'image suivante et passe le diaporama en mode manuel	---	ok				
	l'utilisateur clic sur "lecteur"	Le programme active le mode automatique, passe a l'image suivante automatiquement toutes les 3 secondes	---		Le programme active le mode automatique, passe a l'image suivante automatiquement toutes les 3 secondes	---	ok				
	l'utilisateur clic sur "Pause"	Le programme met le la lecture du diaporama en mode manuel et se stop sur l'image en cours	---		Le programme met le la lecture du diaporama en mode manuel et se stop sur l'image en cours	---	ok				

Version v4–

jeux de test :

V4 jeux de test:

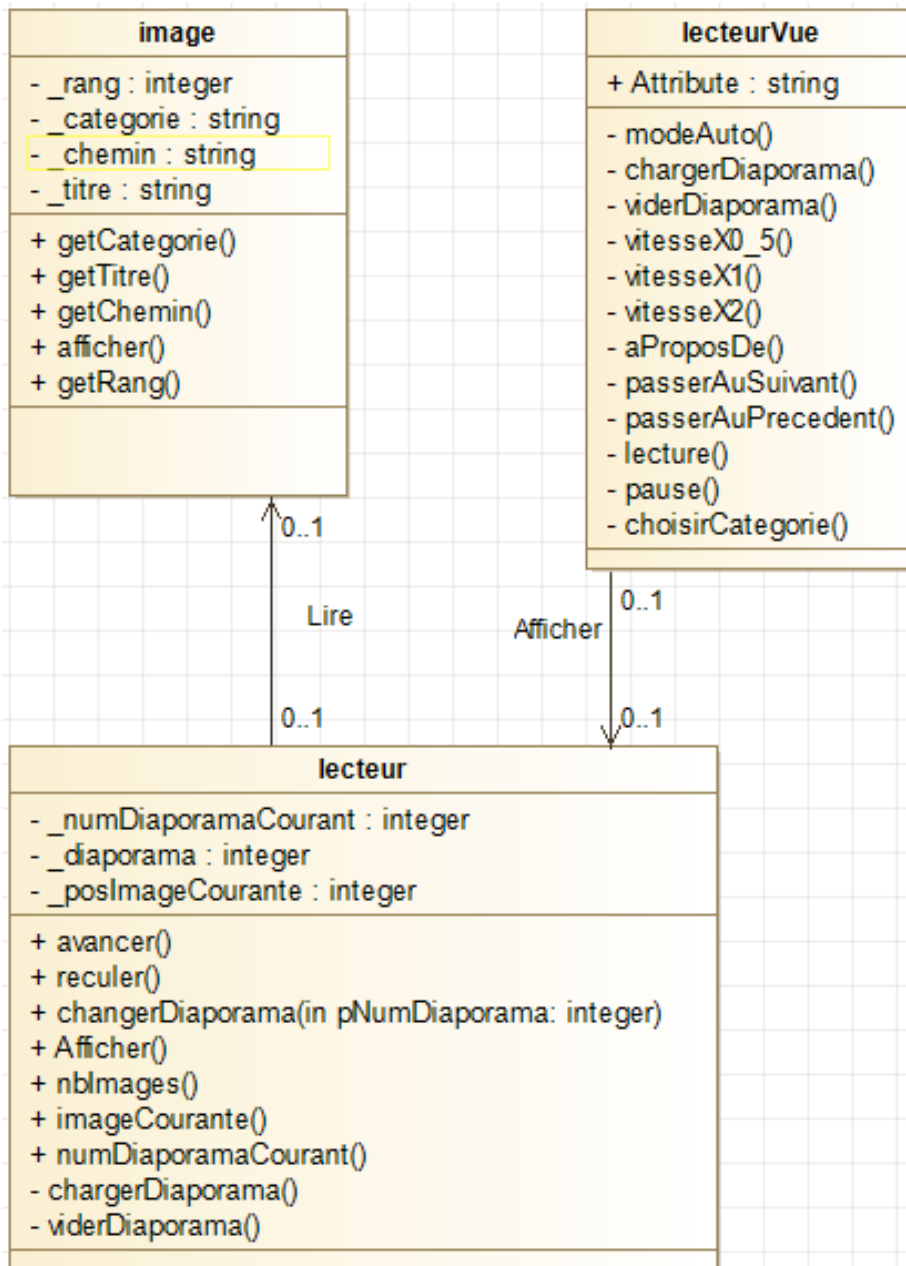
recherchePremiereOccDecEntier

				Réalisation test 1			Résultats test 2		
Description scénario	Résultats attendus			résultats OBTENUS		Remarques : description du problème constaté lors de l'exécution - Idées de solution	résultats OBTENUS		Remarques
	trouvé	pos		trouvé	pos		trouvé	pos	
l'utilisateur clic sur "<"	Le programme passe a l'image précédente et passe le diaporama en mode manuel	---		Le programme passe a l'image précédente et passe le diaporama en mode manuel	---	ok			
l'utilisateur clic sur ">"	Le programme passe a l'image suivante et passe le diaporama en mode manuel	---		Le programme passe a l'image suivante et passe le diaporama en mode manuel	---	ok			
l'utilisateur clic sur "lecteur"	Le programme active le mode automatique, passe a l'image suivante automatiquement toutes les 3 secondes	---		Le programme active le mode automatique, passe a l'image suivante automatiquement toutes les 3 secondes	---	ok			
l'utilisateur clic sur "Pause"	Le programme met le la lecture du diaporama en mode manuel et se stop sur l'image en cours	---		Le programme met le la lecture du diaporama en mode manuel et se stop sur l'image en cours	---	ok			
l'utilisateur clic sur "parametre" puis sur "changer diaporama"	Le programme charge le diaporama et affiche la première image dans l'ordre	---		Le programme charge le diaporama et affiche la première image dans l'ordre.	---	ok			
l'utilisateur clic sur "parametre" puis sur "Enlever diaporama"	Le programme enleve le diaporama charger	---		Le programme enleve le diaporama charger	---	ok			
l'utilisateur clic sur "parametre" puis sur 'vitesse de défilement' puis sur "X0,5" quand le diaporama est en mode automatique	Le programme met la vitesse du diaporama 0,5 fois moins rapide	---		La console affiche "Je change la vitesse a x0.5 "	---	ok			
l'utilisateur clic sur "parametre" puis sur 'vitesse de défilement' puis sur "X1", quand le diaporama est en mode automatique	Le programme met la vitesse du diaporama a la vitesse programmer de base (toutes les 3 secondes)	---		La console affiche "Je change la vitesse a x1 "	---	ok			
l'utilisateur clic sur "parametre" puis sur 'vitesse de défilement' puis sur "X2", quand le diaporama est en mode automatique	Le programme met la vitesse du diaporama 2 fois plus rapide	---		La console affiche "Je change la vitesse a x2 "	---	ok			

Version v5 –

10. Diagramme de classes (UML)

Le diagramme de classe est le même qu'à la version précédente.



11. Comportement de l'application

11.1 Diagramme états-transitions-actions du lecteur de diaporamas (v5)

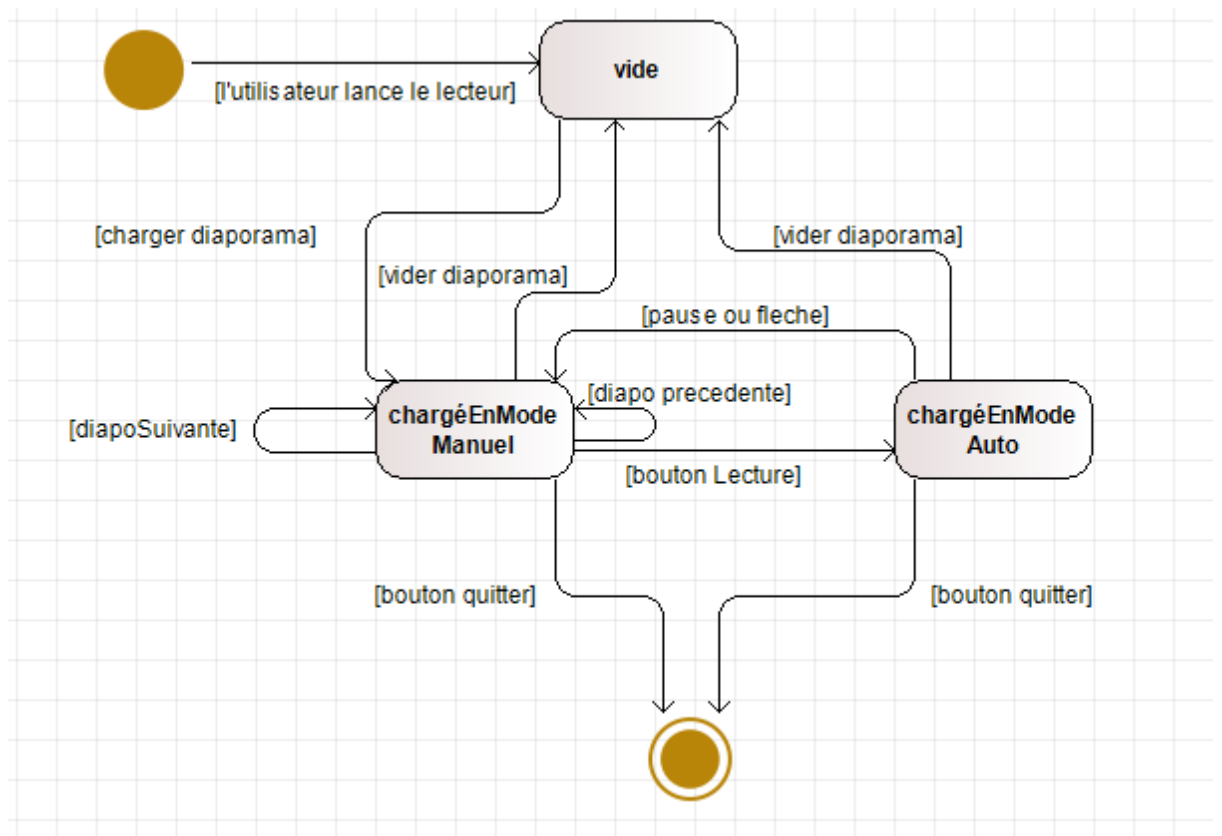


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v5

11.2 Dictionnaire des états, événements et Actions (v5)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
chargéEnModeManue l	le lecteur a un diaporama a lire et l' utilisateur doit passer les diapositives lui même
chargéEnModeAuto	le lecteur a un diaporama a lire et les diapositives se passent toutes seules
vide	le lecteur n'a pas de diaporama a lire

Tableau 2 : États du lecteur de diaporamas – v5

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomEvénement</i>	<i>Signification</i>
lancer le lecteur	permet de lancer dans son état vide
bouton quitter	permet de fermer le lecteur
bouton lecture	met le diapo en mode automatique
diapo suivante	passé à la diapositive suivante en mode manuel et fait basculer en mode manuel si en mode auto
diapo précédente	passé à la diapositive précédente en mode manuel et fait basculer en mode manuel si en mode auto
vider diaporama	permet de vider le lecteur lorsqu'il est chargé
bouton pause	permet d'arrêter le mode auto
charger diaporama	permet de charger un diaporama pour que le lecteur le lise

Tableau 3 : Evénements faisant changer le diaporama d'état – v5

Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>	
bouton lecture	le mode auto se lance, les diapos défilent automatiquement	
diapo suivante	le lecteur passe en mode manuel et la diapositive suivante s'affiche à l'écran	

diapo précédente	le lecteur passe en mode manuel et la diapositive précédente s'affiche à l'écran	
vider diaporama	le lecteur se vide et les boutons deviennent cliquables	
bouton pause	le mode auto s'arrête et la diapositive actuelle reste à l'écran	
charger diaporama	le lecteur se remplit et les boutons deviennent cliquables	

Tableau 4 : Actions à réaliser lors des changements d'état – lecteur de diaporamas v5

11.3 Table T_EtatsEvenementsActions (v5)

Correspondance matricielle du diagramme états-transitions de l'application :

- en ligne : les **états** du lecteur de diaporamas (éventuel état de départ d'une transition)
- en colonne : les **événements** faisant changer le lecteur d'état (déclencheur d'une transition)
- dans chaque cellule : l'état d'arrivée de la transition + action/traitement à faire + éventuellement garde accompagnant la transition

É v é n e m e n t à	l a n c e r l e l e c t e u r	bouton quitte r	bouton pause	bouto n lectu re	diapo précé dente /suiv ante	vider diapo rama	charge r diapor ama
nomEtat							

chargéEnModeManuel	x	etat final	x	chargéEnModeManuel	x	vide	x
chargéEnModeAuto	x	etat final	chargéEnModeManuel	x	chargéEnModeManuel	vide	x
vide	x	etat final	x	x	x	x	chargéEnModeManuel

Tableau 5 : Matrice d'états-transitions du lecteur de diaporamas – v5

L'intérêt de cette vue matricielle est qu'elle permet une préparation naturelle et aisée de l'étape suivante de programmation.

12. Implémentation et tests

12.1 Implémentation (v5)

Liste et rôle des fichiers de cette version :

lecteurVue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas. permet de mettre en place les signaux de chaque boutons au action qui leurs sont attribués.
lecteurVue.cpp	Corps de la classe LecteurVue
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner

lecteur.h	Spécification de la classe Lecteur. permet l'affichage et de gérer les différents état du diaporama
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image. Permet de récupérer toutes les informations sur les images
image.cpp	Corps de la classe Image
main.cpp	exécution et affichage du programme

Remarques sur l'implémentation :

Commenter brièvement les choix importants d'implémentation réalisés, **comme par exemple, les signals/slots**

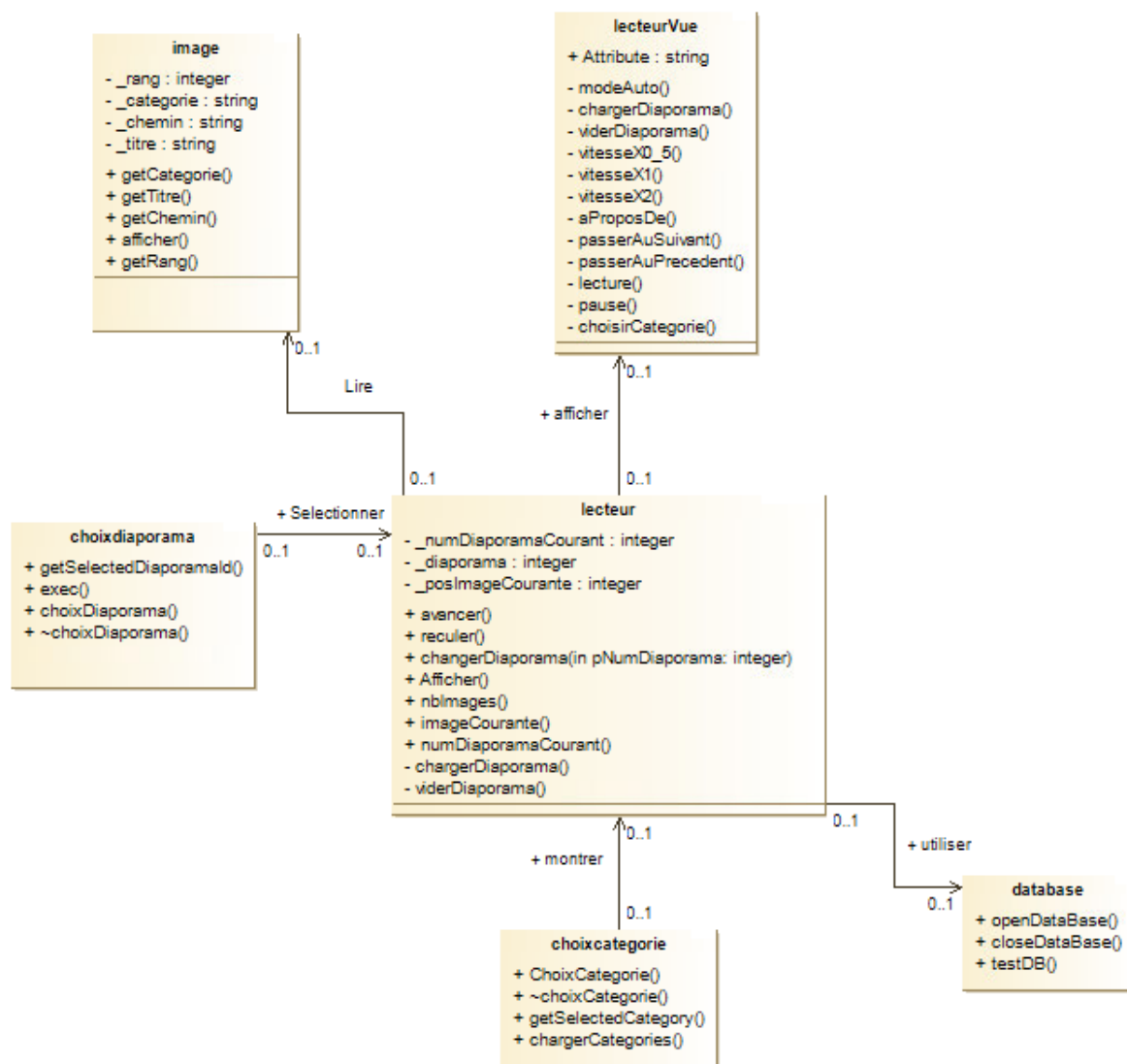
12.2 Tests (v5)

jeux de test :

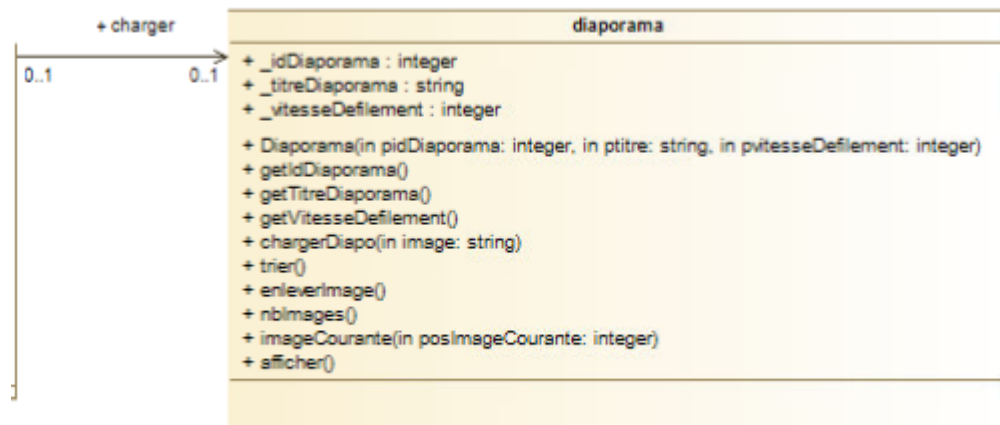
V6 jeux de test:																			
recherchePremiereOccDecEntier																			
					Réalisation test 1						Résultats test 2								
					résultats OBTENUS				Remarques : description du problème constaté lors de l'exécution - Idées de solution	résultats OBTENUS									
					trouvé		pos			trouvé		pos							
Description scénario					trouvé		pos												
Execution de la requete permetant de charger les différents diaporamas					Le programme afficher les image tiré de la base de donnée grace a la requête sql		---		ok										
Affichage des différentes diaporama.					Affichage du nom de l'image tiré de la base de donnée		---		ok										
L'utilisateur demande a charger un diaporama					affichage de la fenêtre contenant la premiere image du diaporama et toutes les information et les boutons permettant de poursuivre le diaporama.		---		ok										

Version v7–

Diagramme de classe (UML):



Pour un manque de place nous n'avons pas représenté la classe diaporama mais elle est toujours présente.



jeux de test :

V7 jeux de test:								
recherchePremiereOccDecEntier					Réalisation test 1			
	Description scénario	Résultats attendus			résultats OBTENUS		problème constaté lors de l'exécution -	
		trouvé	pos		trouvé	pos	Idées de solution	
	Affichage des différentes diaporama des base de donnée.	Le programme affiche les diaporamas présente dans les bases de données via une fenêtre	---		Le programme affiche les diaporamas présente dans les bases de données via une fenêtre	---	ok	
	l'utilisateur selectionne l'un diaporamas	Affichage dans le programme la premieres image du diaporama selectionner avec les inforamtions du diaporama (nom, rangs etc..)	---		Affichage dans le programme la premieres image du diaporama selectionner avec les inforamtions du diaporama (nom, rangs etc..)	---	ok	
	L'utilisateur demande a trier par catégorie	affichage de la fenêtre contenant toutes les catégorie du diaporama actuelle et attend la selection de l'utilisateur pour le choix de la catégorie, une fois selectionner le programme afficher toutes les image du diaporame avec la catégorie voulu	---		affichage de la fenêtre contenant toutes les catégorie du diaporama actuelle et attend la selection de l'utilisateur pour le choix de la catégorie, une fois selectionner le programme afficher toutes les image du diaporame avec la catégorie voulu	---	ok	

13. Bilan

Dépôt Git où trouver le projet complet (les versions réalisées)

<https://github.com/Niwowo/LecteurDiaporama.git>

Temps global de travail (pour le groupe)

Nous avons réalisé un travail équitable sur l'ensemble du groupe, nous comptons une vingtaine d'heure pour chaque membre du groupe. Le temps de travail a été constant tout au long de la SAE.

Apprentissages majeurs

Le développement d'application ne peut se faire sans pratiquer réellement, aller à la rencontre de tous les petits problèmes que l'utilisation d'une base de données amène est essentielle

Difficultés majeures

La connexion et l'utilisation de la base de données a été plutôt difficile et la compréhension générale de chaque version nous a coûté plutôt cher.

Points positifs / négatifs de l'activité

Le sujet n'est pas clair pour le travail à faire sur le rendu, pour chaque version nous ne savons pas ce qu'il faut réaliser précisément. Le point positif de cette SAE est le suivi et la réaction par la professeur est appréciable.