



S2.01

DEVELOPPEMENT D'UNE APPLICATION

Lecteur de diaporamas

Dossier d'analyse et conception

BUT Informatique Semestre 2

IUT de Bayonne et du Pays Basque

Département Informatique

DUMOLIE ALEXIS TD3/TP5
MORENO ALEX TD3/TP5

Année universitaire
2022-2023

Rendu le 05/06/2023

TABLE DES MATIÈRES

1. SCÉNARIOS.....	3
2. DIAGRAMME DE CLASSEES (UML).....	4
Version v0 – Version console seule	6
1. Implémentation et tests	6
1.1 Implémentation.....	6
1.2 Test.....	6
Version v1 – projet Graphique seul.....	8
1. Éléments d’interface	8
2. Implémentation et tests	9
2.1 Test.....	10
Version v2 –	11
1. Diagramme de classes (UML)	11
2. Comportement de l’application	12
2.1 Diagramme états-transitions-actions du lecteur de diaporamas (v2).....	12
2.2 Dictionnaire des états, événements et Actions (v2).....	13
2.3 Table T_EtatsEvenementsActions (v2).....	14
3. Implémentation et tests	15
3.1 Implémentation (v2).....	15
Version v3 –	17
1. Diagramme de classes (UML)	17
2. Comportement de l’application	17
2.1 Dictionnaire des états, événements et Actions (v3).....	18
2.2 Table T_EtatsEvenementsActions (v3).....	20
3. Implémentation et tests	21
3.1 Implémentation (v3).....	21
3.2 Tests (v3).....	21
Version v4 –	23
1. Diagramme de classes (UML)	23
2. Comportement de l’application	23
Diagramme états-transitions-actions du lecteur de diaporamas (v4).....	23
2.1 Dictionnaire des états, événements et Actions (v4).....	23
2.2 Table T_EtatsEvenementsActions (v4).....	26
3. Implémentation et tests	27
3.1 Implémentation (v4).....	27
3.2 Tests (v4).....	27
Version v5 –	30
1. Diagramme de classes (UML)	30
2. Comportement de l’application	30
2.1 Diagramme états-transitions-actions du lecteur de diaporamas (v5).....	30
2.2 Dictionnaire des états, événements et Actions (v5).....	30
3. Implémentation et tests	30
3.1 Implémentation (v5).....	30
3.2 Tests (v5).....	31
Bilan.....	32

1. SCÉNARIOS

Scénario nominal “mode auto”

Acteur	Système
1. L'utilisateur choisi le mode auto	
	2. Le système affiche la première image.
	3. Le système affiche l'image suivante toute les deux secondes

Scénario alternatif

Acteur	Système
	3. Le système affiche l'image suivante toute les deux secondes
3.A.1 L'utilisateur relance le mode auto	
	retour au 2

Acteur	Système
	3. Le système affiche l'image suivante toute les deux secondes
3.B L'utilisateur choisi d'arrêter le diaporama	
	3.B.2 Le système laisse afficher l'image courante

Scénario nominal “mode manuel”

Acteur	Système
1. L'utilisateur choisi de passer à l'image suivante	
	2. Le système affiche l'image suivante

Acteur	Système
3. L'utilisateur choisi de passer à l'image précédente	
	4. Le système affiche l'image précédente

Scénario alternatif

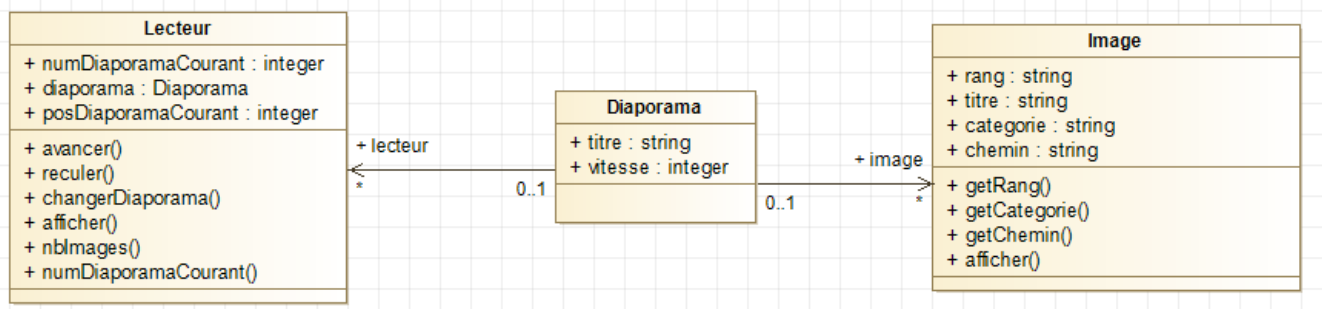
Acteur	Système
--------	---------

	1. Le système affiche la dernière image
2. L'utilisateur choisi de passer à l'image suivante	
	3. Le système affiche la première image

Acteur	Système
	3. Le système affiche la première image
4. L'utilisateur choisi de passer à l'image précédente	
	3. Le système affiche la dernière image

2. DIAGRAMME DE CLASSEES (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.



- (b) Dictionnaire des éléments pour chaque classe

Classe Lecteur			
Nom attribut	Signification	Type	Exemple
numDiaporamaCourant	Numéro du diaporama courant	unsigned	1
diaporama	Diaporama courant	Diaporama	
posImageCourante	Position de l'image courante dans le diaporama	unsigned int	2

Classe Image			
Nom attribut	Signification	Type	Exemple
rang	Le rang de l'image dans le diaporama	unsigned int	3
categorie	La catégorie de l'image	string	"personne"
titre	Le titre de l'image	string	"Blanche Neige"

chemin	Le chemin d'accès de l'image	string	"C:\\carte sDisney\\c arteDisney 2.gif"
--------	------------------------------	--------	--------------------------------------------------

(c) Dictionnaire des méthodes :

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
```

Capture de la classe Image

```
#ifndef IMAGE_H
#define IMAGE_H
#include <iostream>
using namespace std;

class Image
{
public:
    Image(unsigned int pRang=0,
           string pCategorie="", string pTitre="", string pChemin = "");
    unsigned int getRang();
    string getCategorie();
    string getTitre();
    string getChemin();
    void afficher(); // affiche tous les champs de l'image

private:
    unsigned int _rang; // rang de l'image au sein du diaporama
    // auquel l'image est associée */
    string _titre; // intitulé de l'image
    string _categorie; // catégorie de l'image (personne, animal, objet)
    string _chemin; // chemin complet vers le dossier où se trouve l'image
};

#endif // IMAGE_H
```

Version v0 – Version console seule

1. Implémentation et tests

1.1 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	Teste les méthodes de la classe Lecteur

1.2 Test

Test avec le programme fourni main.cpp

```
Lecteur vide
Diaporama num. 1 selectionne.
4 images chargees dans le diaporama
Diaporama num. 1
image courante : image( rang:1, titre:Grincheux, categorie:personne, chemin:C:\cartesDisney\carteDisney1.gif)

Test avancer() : 4 fois
avancer() :
Diaporama num. 1
image courante : image( rang:2, titre:Cendrillon, categorie:personne, chemin:C:\cartesDisney\carteDisney4.gif)
avancer() :
Diaporama num. 1
image courante : image( rang:3, titre:Blanche Neige, categorie:personne, chemin:C:\cartesDisney\carteDisney2.gif)
avancer() :
Diaporama num. 1
image courante : image( rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)
avancer() :
Diaporama num. 1
image courante : image( rang:1, titre:Grincheux, categorie:personne, chemin:C:\cartesDisney\carteDisney1.gif)

Test reculer() : 5 fois
reculer() :
Diaporama num. 1
image courante : image( rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)
reculer() :
Diaporama num. 1
image courante : image( rang:3, titre:Blanche Neige, categorie:personne, chemin:C:\cartesDisney\carteDisney2.gif)
reculer() :
Diaporama num. 1
image courante : image( rang:2, titre:Cendrillon, categorie:personne, chemin:C:\cartesDisney\carteDisney4.gif)
reculer() :
Diaporama num. 1
image courante : image( rang:1, titre:Grincheux, categorie:personne, chemin:C:\cartesDisney\carteDisney1.gif)
reculer() :
Diaporama num. 1
image courante : image( rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)

Enlever le diaporama courant = Choisir diaporama 0
0 images restantes dans le diaporama.
Lecteur vide
```

Capture du résultat en console obtenu :

```
ima [main]
rama.pro

1  #include "lecteur.h"
2
3  Lecteur::Lecteur()
4  {

Sortie de l'application
untitled x lecteurDiaporama x
16:20:37: Starting F:\IUT\Annee_1\S2\SAE\S2.01DevApp\S201DevApp\v0_version_console_pourEtudiants\build-lecteurDiaporama
Lecteur vide
Diaporama num. 1 selectionne.
4 images chargees dans le diaporama
Diaporama num. 1
image courante : image( rang:1, titre:Grincheux, categorie:personne, chemin:C:\cartesDisney\carteDisney1.gif)

Test avancer() : 4 fois
avancer() :
Diaporama num. 1
image courante : image( rang:2, titre:Cendrillon, categorie:personne, chemin:C:\cartesDisney\carteDisney4.gif)

avancer() :
Diaporama num. 1
image courante : image( rang:3, titre:Blanche Neige, categorie:personne, chemin:C:\cartesDisney\carteDisney2.gif)

avancer() :
Diaporama num. 1
image courante : image( rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)

avancer() :
Diaporama num. 1
image courante : image( rang:1, titre:Grincheux, categorie:personne, chemin:C:\cartesDisney\carteDisney1.gif)

Test reculer() : 5 fois
reculer() :
Diaporama num. 1
image courante : image( rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)

reculer() :
Diaporama num. 1
image courante : image( rang:3, titre:Blanche Neige, categorie:personne, chemin:C:\cartesDisney\carteDisney2.gif)

reculer() :
Diaporama num. 1
image courante : image( rang:2, titre:Cendrillon, categorie:personne, chemin:C:\cartesDisney\carteDisney4.gif)

reculer() :
Diaporama num. 1
image courante : image( rang:1, titre:Grincheux, categorie:personne, chemin:C:\cartesDisney\carteDisney1.gif)

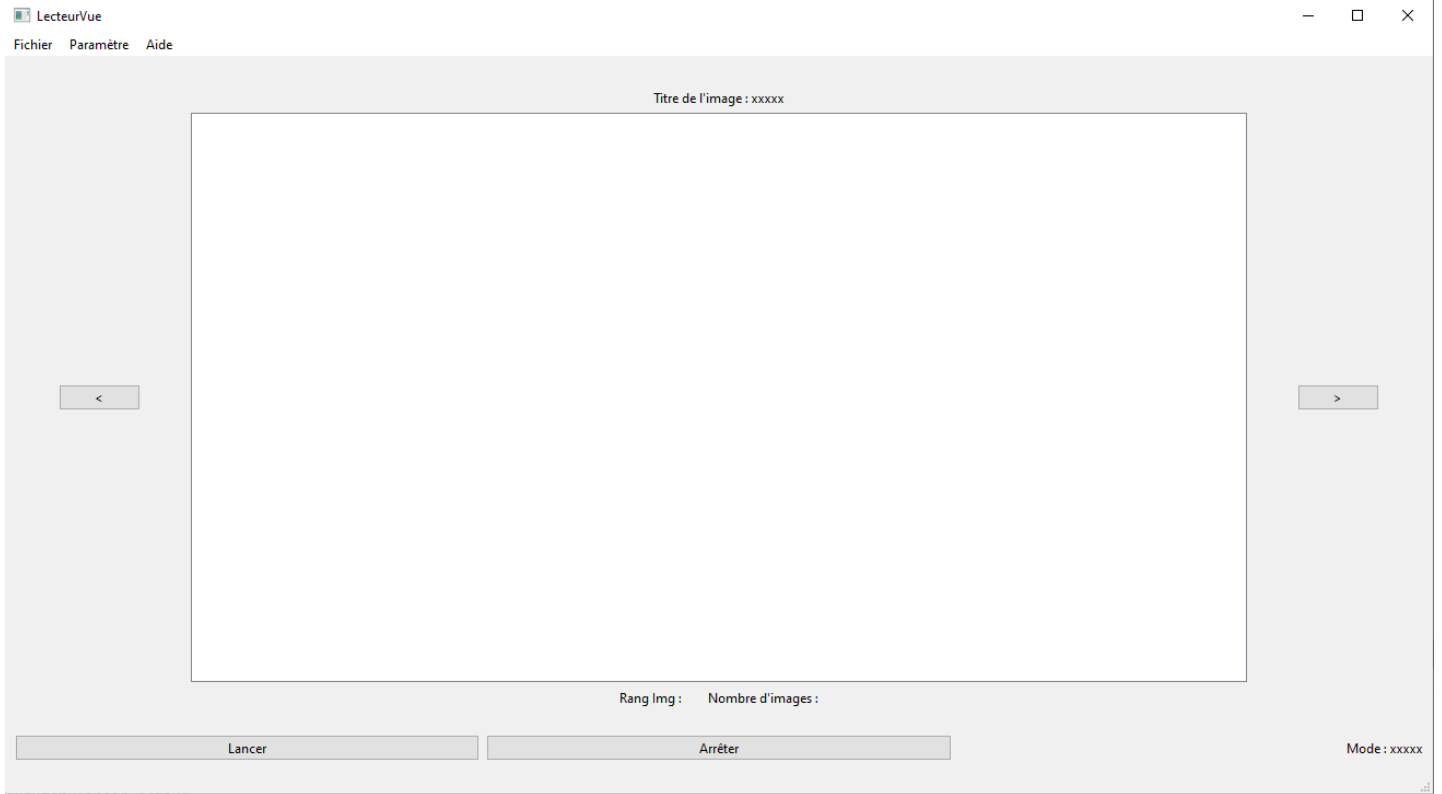
reculer() :
Diaporama num. 1
image courante : image( rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)

Enlever le diaporama courant = Choisir diaporama 0
0 images restantes dans le diaporama.
Lecteur vide
```

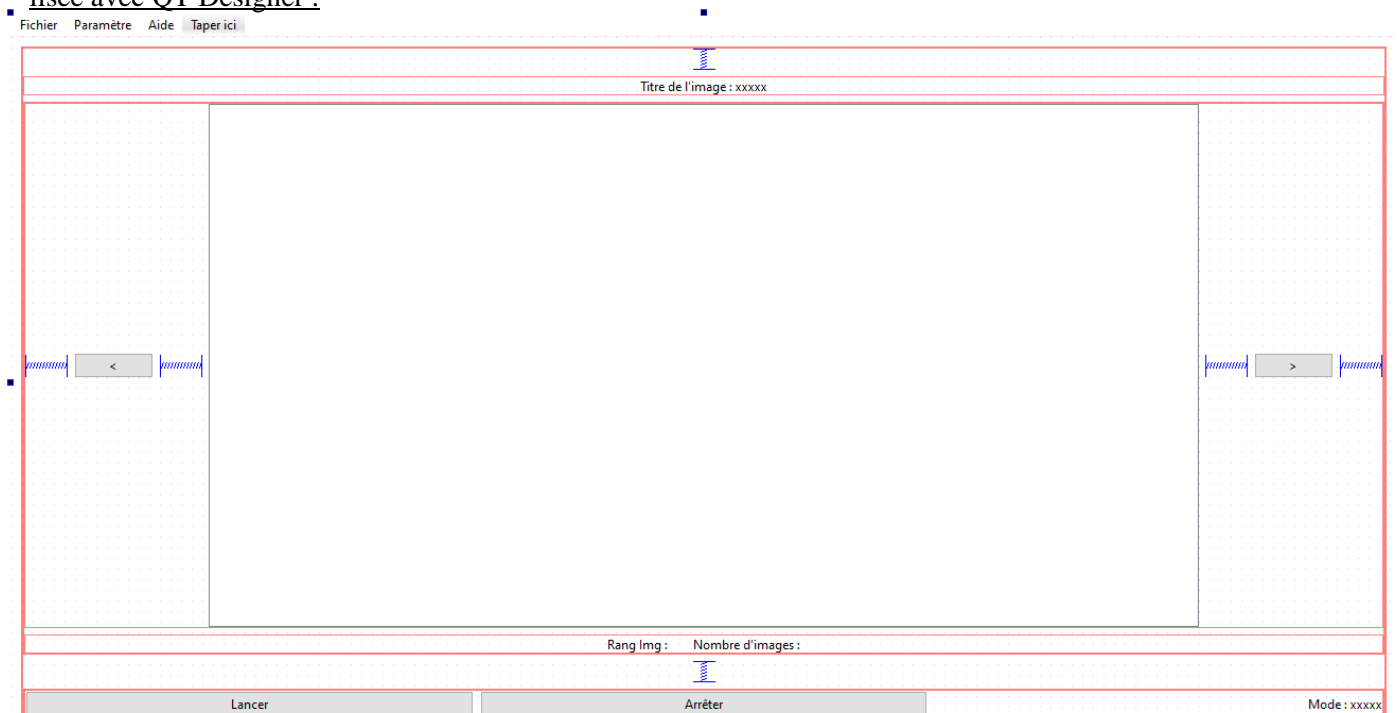
Version v1 – projet Graphique seul

1. Éléments d'interface

Interface finale générée avec QT :



Interface graphique réa-
lisée avec QT Designer :



Les layouts :

Objet	Classe
▼ LecteurVue	QMainWindow
▼ menubar	QMenuBar
menuAide	QMenu
menuParam_tre	QMenu
menuTitre_diaporama	QMenu
statusbar	QStatusBar
▼ widget	QWidget
▼ horizontalLayout_5	QHBoxLayout
▼ verticalLayout_5	QVBoxLayout
▼ horizontalLayout	QHBoxLayout
titreImage	QLabel
▼ verticalLayout	QVBoxLayout
▼ horizontalLayout_2	QHBoxLayout
btnPrecedent	QPushButton
btnSuivant	QPushButton
horizontalSpacer	Spacer
horizontalSpacer_2	Spacer
horizontalSpacer_3	Spacer
horizontalSpacer_4	Spacer
image	QGraphicsView
▼ verticalLayout_2	QVBoxLayout
infoDiapo	QLabel
▼ verticalLayout_3	QVBoxLayout
▼ horizontalLayout_3	QHBoxLayout
btnArret	QPushButton
btnLancer	QPushButton
mode	QLabel
verticalSpacer	Spacer
verticalSpacer_3	Spacer

2. Implémentation et tests

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	Teste les méthodes de la classe Lecteur

Remarques sur l'implémentation :

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

Nous avons connecté le signal "clicked()" du bouton suivant au slot "boutonSuivant()", ainsi un clic du bouton suivant appelle la fonction boutonSuivant().

Nous avons connecté le signal "clicked()" du bouton précédent au slot "boutonPrecedent()", ainsi un clic du bouton précédent appelle la fonction boutonPrecedent().

Nous avons connecté le signal "clicked()" du bouton lancer au slot "boutonLancer()", ainsi un clic du bouton

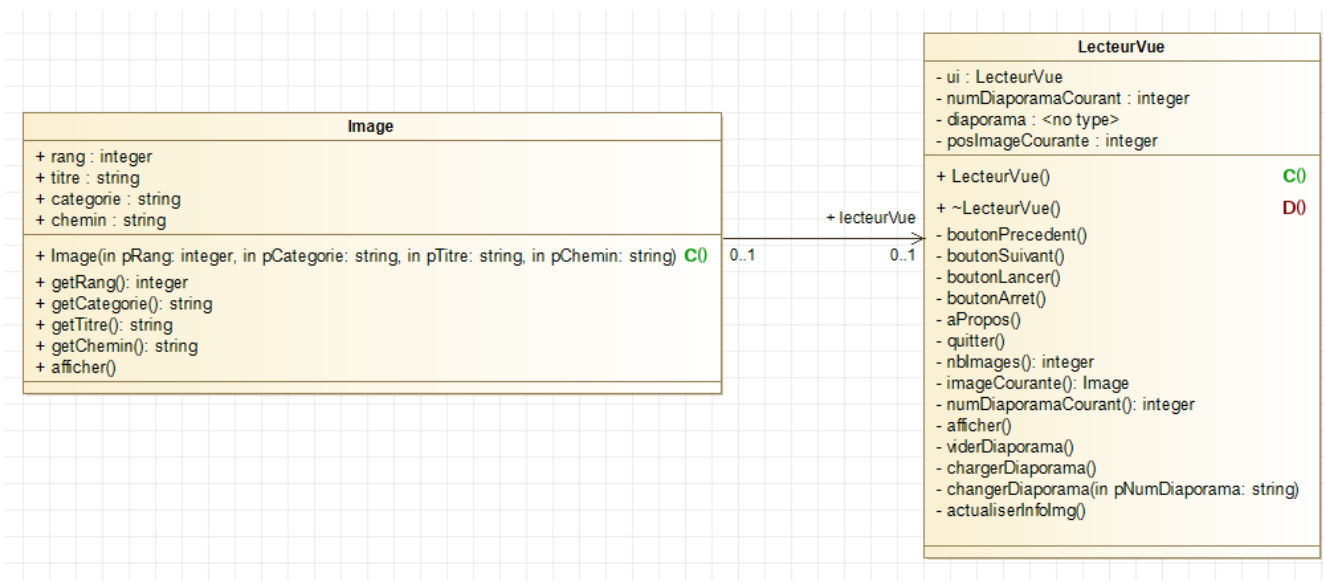
lancer appelle la fonction boutonLancer().

Nous avons connecté le signal "clicked()" du bouton arrêter au slot "boutonArreter()", ainsi un clic du bouton arrêter appelle la fonction boutonArreter().

2.1 Test

Action réalisée	Réponse attendue	Réponse obtenue	Preuve
clic du bouton suivant	"Bouton suivant activé"	"Bouton suivant activé"	16:14:33: Starting F:\ Bouton suivant activé
clic du bouton précédent	"Bouton précédent activé"	"Bouton précédent activé"	16:14:33: Starting F:\IUT Bouton suivant activé Bouton precedent activé
clic du bouton arrêter	"Bouton arrêt activé"	"Bouton arrêt activé"	
clic du bouton lancer	"Bouton lancer activé"	"Bouton lancer activé"	16:14:33: Starting F:\I Bouton suivant activé Bouton precedent activé Bouton arrêt activé Bouton lancer activé

1. Diagramme de classes (UML)



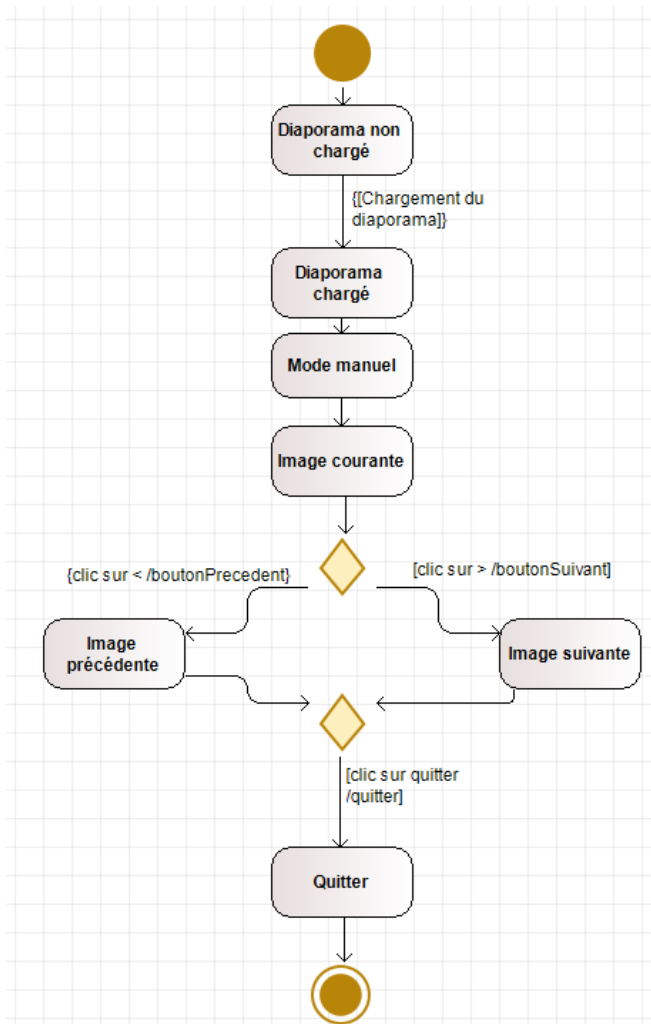
Dictionnaire des éléments pour chaque classe :

Classe Image			
Nom attribut	Signification	Type	Exemple
rang	Le rang de l'image dans le diaporama	unsigned int	3
categorie	La catégorie de l'image	string	"personne"
titre	Le titre de l'image	string	"Blanche Neige"
chemin	Le chemin d'accès de l'image	string	"C:\\cartesDisney\\carteDisney2.gif"

Classe Image			
Nom attribut	Signification	Type	Exemple
ui	Le rang de l'image dans le diaporama	LecteurVue	x
numDiaporamaCourant	Numéro du diaporama dourant	unsigned int	1
diaporama	Pointeur vers les images du diaporama	Diaporama	x
posImageCourante	Rang courant de l'image	unsigned int	2
etatDiapo	Etat du diapo soit automatique soit manuel	string	Automatique

2. Comportement de l'application

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



2.2 Dictionnaire des états, événements et Actions (v2)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
Diaporama non chargé	Lorsque qu'on lance l'application le diaporama n'est pas chargé
Diaporama chargé	L'application charge le contenu du diaporama
Mode manuel	Le mode de fonctionnement par défaut
Image courante	Image courante affichée
Image suivante	Image suivante affichée
Image précédente	Image précédente affichée
Quitter	L'utilisateur décide de quitter

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomÉvénement</i>	<i>Signification</i>
Ouverture de l'application	L'utilisateur lance l'application
L'application charge le diaporama	Dans cette version, l'utilisateur ne doit pas charger le diaporama, c'est l'application qui le fait automatiquement
Mode manuel par défaut	Idem ici, l'utilisateur n'a rien à changer, le mode manuel est le seul dans cette version
Clic sur précédent	Lorsque l'utilisateur clique sur précédent, l'application affiche l'image précédente
Clic sur suivant	Lorsque l'utilisateur clique sur suivant, l'application affiche l'image suivante
Clic sur quitter	L'utilisateur choisit de quitter l'application, le lecteur se ferme.

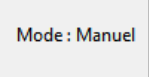
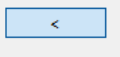
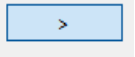
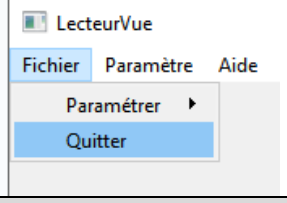
Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>
Chargement du diaporama	L'application charge le diaporama
Passage à l'image précédente	L'utilisateur clique et fait changer l'image vers l'image précédente
Passage à l'image suivante	L'utilisateur clique et fait changer l'image vers l'image suivante
Quitter l'application	L'utilisateur clique sur quitter le diaporama et l'application se ferme

2.3 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</i> ▮	X	X				
<i>Événement</i> ▮ <i>nomEtat</i>	Ouverture de l'application	L'application charge le diaporama	Mode manuel par défaut	Clic sur précédent	Clic sur suivant	Clic sur quitter
Chargement du diaporama		Diaporama se charge				
Passage à l'image précédente				Passe à l'image précédente		
Passage à l'image suivante					Passe à l'image suivante	
Quitter l'application						L'application se ferme

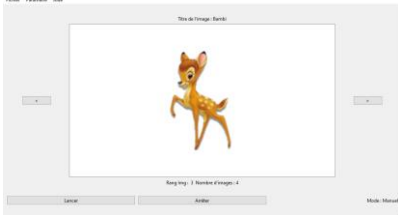
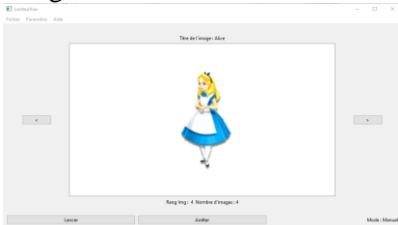

3. Implémentation et tests

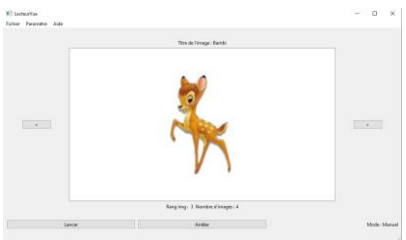
3.1 Implémentation (v2)

Liste et rôle des fichiers de cette version :

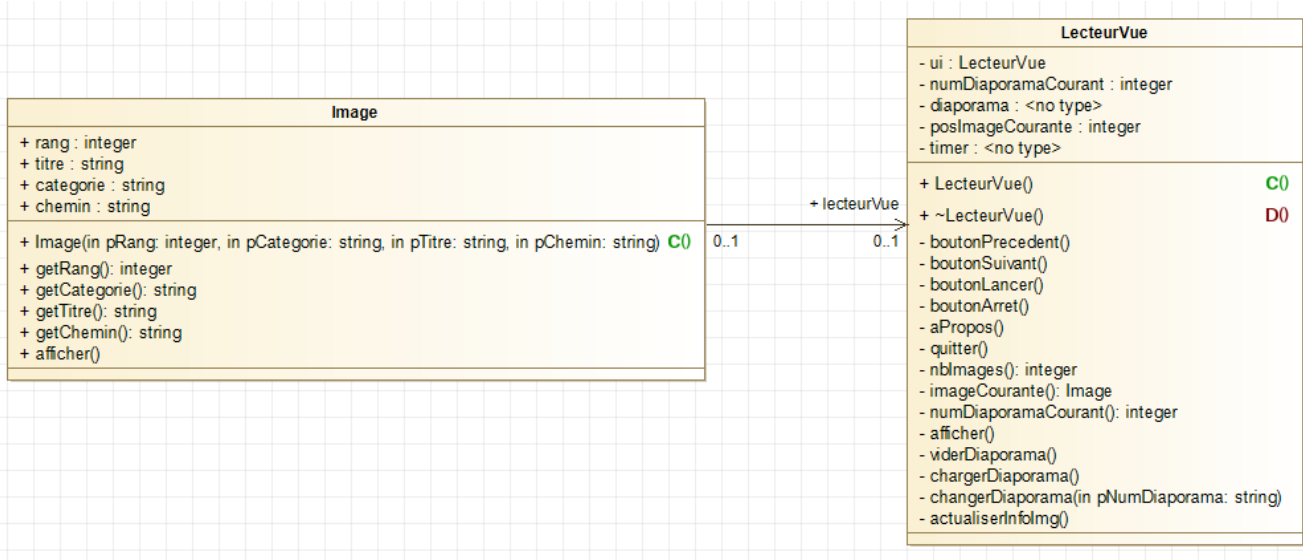
lecteurVue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas LecteurVue est la fenêtre principale qui contient toutes les fonctions principales associées aux actions du diaporama
lecteurVue.cpp	Corps de la classe LecteurVue.
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner
image.h	Spécification de la classe Image La classe image contient des variables membres pour stocker les informations spécifiques à l'image, ainsi que des fonctions pour accéder à ces informations et les afficher.
image.cpp	Corps de la classe Image
main.cpp	Permet de créer la fenêtre principale et de l'afficher

3.2 Tests (v2)

Action réalisée	Réponse attendue	Réponse obtenue	Preuve
clic du bouton suivant	Image suivante	Image suivante	<p>Image 3</p>  <p>Appui suivant</p> <p>Image 4</p> 
clic du bouton précédent	Image précédente	Image précédente	<p>Image 4</p>  <p>Appui précédent</p> <p>Image 3</p>

			
clic du bouton arrêter	“Bouton arrêt activé”	“Bouton arrêt activé”	16:52:24: Starting F:\IUT\Anne Diaporama num. 1 selectionne. 4 images chargees dans le diap Bouton arrêt activé
clic du bouton lancer	“Bouton lancer activé”	“Bouton lancer activé”	16:51:41: Starting F:\IUT\Anne Diaporama num. 1 selectionne. 4 images chargees dans le diap Bouton lancer activé

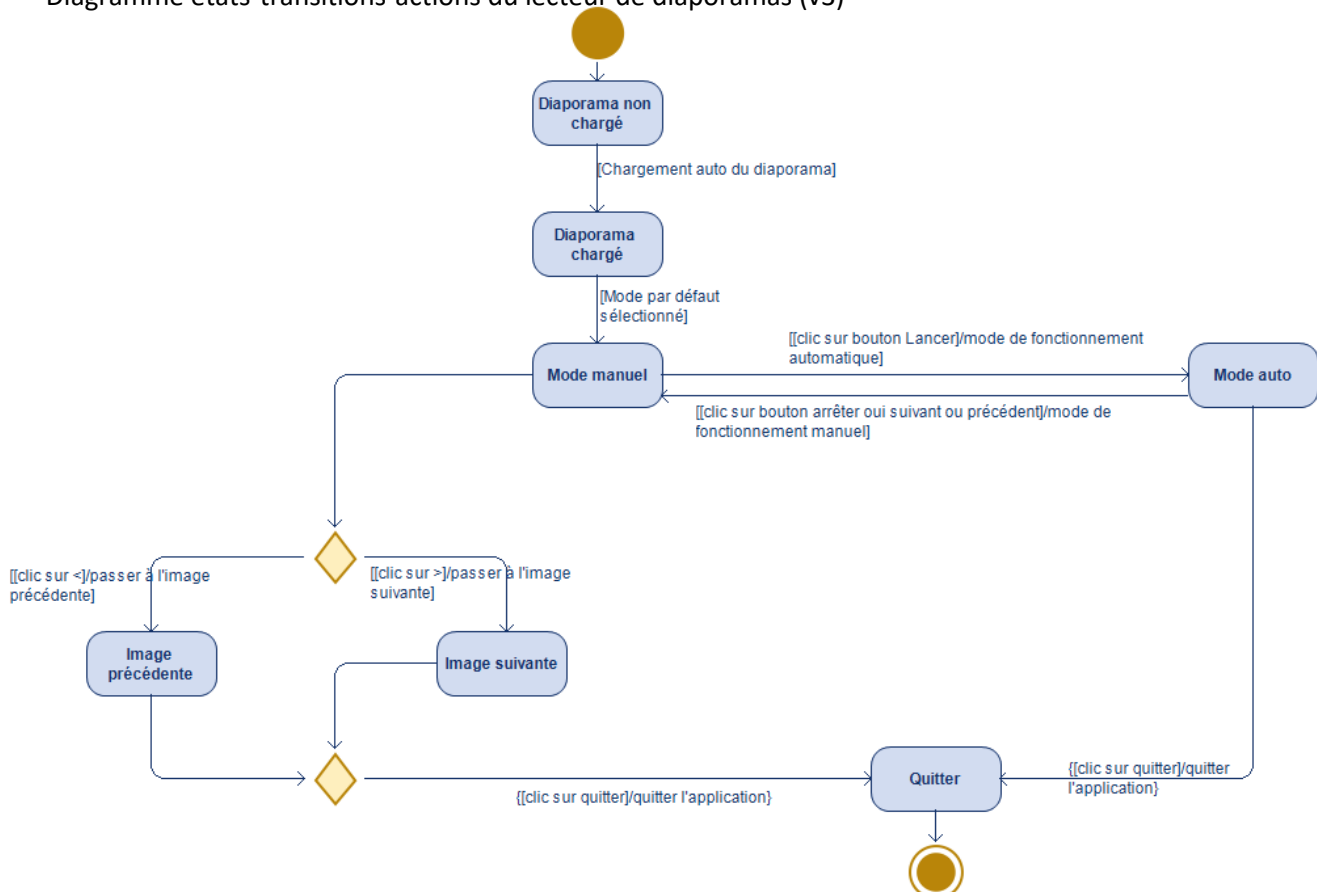
1. Diagramme de classes (UML)



Ajout d'un timer dans la classe LecteurVue qui est de type QTimer (pas possible de le mettre sur modelio) qui permet le défilement automatique du diapo

2. Comportement de l'application

Diagramme états-transitions-actions du lecteur de diaporamas (v3)



2.1 Dictionnaire des états, événements et Actions (v3)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
Diaporama non chargé	Lorsque qu'on lance l'application, le diaporama n'est pas chargé
Diaporama chargé	L'application charge le contenu du diaporama
Mode manuel	Le mode de fonctionnement par défaut
Mode auto	Mode sélectionné par le joueur où les images se défilent automatiquement
Image suivante	Image suivante affichée
Image précédente	Image précédente affichée
Quitter	L'utilisateur décide de quitter

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomEvénement</i>	<i>Signification</i>
Ouverture de l'application	L'utilisateur lance l'application
L'application charge le diaporama	Dans cette version, l'utilisateur ne doit pas charger le diaporama, c'est l'application qui le fait automatiquement
Mode manuel par défaut	Idem ici, l'utilisateur n'a rien à changer, le mode manuel est le seul dans cette version
Clic sur précédent	Lorsque l'utilisateur clique sur précédent, l'application affiche l'image précédente. Fait aussi arrêter le défilement si mode = auto
Clic sûr suivant	Lorsque l'utilisateur clique sur suivant, l'application affiche l'image suivante. Fait aussi arrêter le défilement si mode = auto
Clic sur quitter	L'utilisateur choisit de quitter l'application, le lecteur se ferme.
Clic sur bouton lancer	Le mode auto se lance
Clic sur arrêter	Mode de fonctionnement manuel

Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>
Chargement auto du diaporama	L'application charge le diaporama
Mode par défaut sélectionné	Le mode par défaut est le mode manuel
mode de fonctionnement manuel	Le mode passe à manuel
mode de fonctionnement automatique	Le mode passe à auto
passer à l'image précédente	L'image précédente s'affiche
passer à l'image suivante	L'image suivante s'affiche
quitter l'application	L'application se ferme

2.2 Table T_EtatsEvenementsActions (v3)

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

Élément graphique pregnant en charge cet événement □	X	X	X	Bouton <	Bouton >	Fichier >> Quitter	Bouton Lancer	Bouton arrêter
Événement □ nomEtat	Ouverture de l'application	L'application charge le diaporama	Mode manuel par défaut	Clic sur précédent	Clic sur suivant	Clic sur quitter	Clic sur lancer	Clic sur arrêter
Diaporama non chargé		Chargement du diaporama						
Diaporama chargé								
Mode manuel			Mode par défaut				Passage au mode auto	
Mode auto				Retour au mode manuel	Retour au mode manuel	Quitter l'application		Retour au mode manuel
Image précédente				Passer à l'image précédente				
Image suivante					Passer à l'image suivante			
Quitter						Quitter l'application		

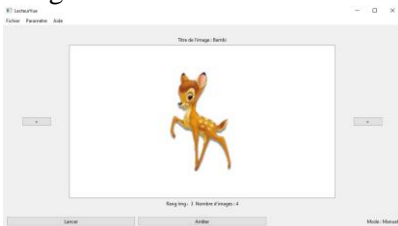


3. Implémentation et tests

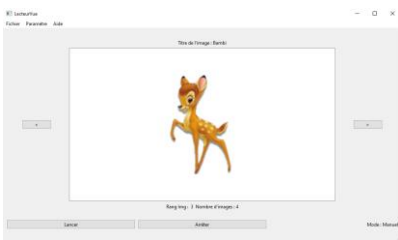
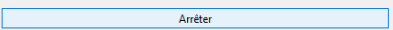
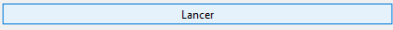
3.1 Implémentation (v3)

Liste et rôle des fichiers de cette version :

lecteurVue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas ROLE IDEM A V2
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 ROLE IDEM A V2
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image ROLE IDEM A V2
image.cpp	Corps de la classe Image
main.cpp	ROLE IDEM A V2

3.2 Tests (v3)

Action réalisée	Réponse attendue	Réponse obtenue	Preuve
clic du bouton suivant	Image suivante	Image suivante	<p>Image 3</p>  <p>Appui suivant</p> <p>Image 4</p> 
clic du bouton précédent	Image précédente	Image précédente	<p>Image 4</p>  <p>Appui précédent</p> <p>Image 3</p>

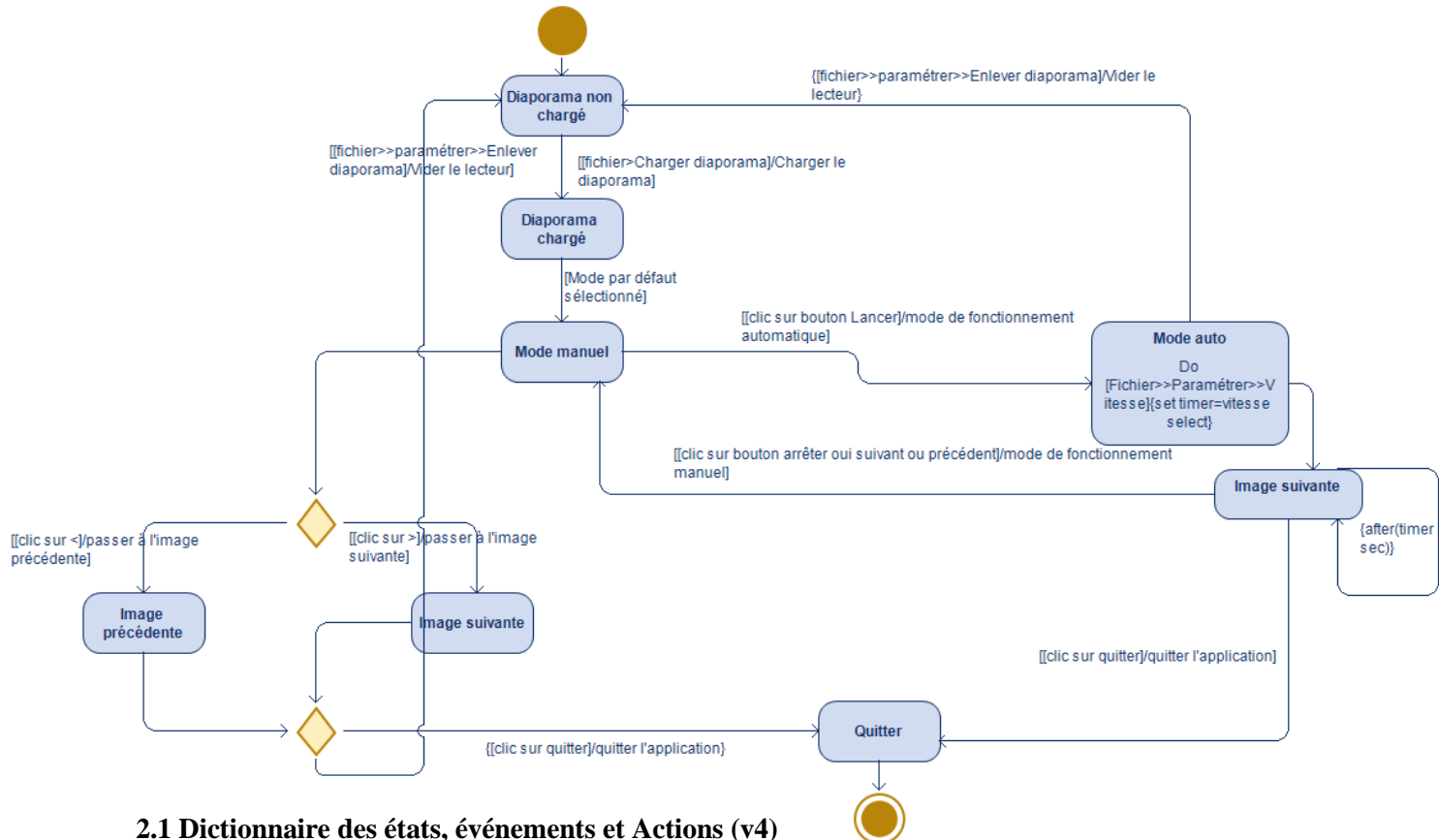
			
clic du bouton arrêter	Passe au mode manuel	Passe au mode manuel	<p>Etat Diapo :</p> <p>Mode : Automatique</p> <p>Clic sur bouton arrêter :</p>  <p>Etat Diapo :</p> <p>Mode : Manuel</p>
clic du bouton lancer	Passe au mode automatique	Passe au mode automatique	<p>Etat Diapo :</p> <p>Mode : Manuel</p> <p>Clic sur bouton lancer:</p>  <p>Etat Diapo :</p> <p>Mode : Automatique</p>

1. Diagramme de classes (UML)

IDEM A V3

2. Comportement de l'application

Diagramme états-transitions-actions du lecteur de diaporamas (v4)



2.1 Dictionnaire des états, événements et Actions (v4)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
Diaporama non chargé	Lorsque qu'on lance l'application, le diaporama n'est pas chargé
Diaporama chargé	L'application charge le contenu du diaporama
Mode manuel	Le mode de fonctionnement par défaut

Mode auto	Mode sélectionné par le joueur où les images se défilent automatiquement
Image suivante	Après timer sec(timer étant une variable contenant la vitesse) l'image passe à la suivante
Image suivante	Image suivante affichée dans le mode manuel
Image précédente	Image précédente affichée
Quitter	L'utilisateur décide de quitter

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomEvénement</i>	<i>Signification</i>
Ouverture de l'application	L'utilisateur lance l'application
L'utilisateur charge le diaporama	L'utilisateur charge manuellement le diaporama
Mode manuel par défaut	Idem ici, l'utilisateur n'a rien à changer, le mode manuel est le seul dans cette version
Clic sur précédent	Lorsque l'utilisateur clique sur précédent, l'application affiche l'image précédente. Fait aussi arrêter le défilement si mode = auto
Clic sûr suivant	Lorsque l'utilisateur clique sur suivant, l'application affiche l'image suivante. Fait aussi arrêter le défilement si mode = auto
Clic sur quitter	L'utilisateur choisit de quitter l'application, le lecteur se ferme.
Clic sur bouton lancer	Le mode auto se lance
Clic sur arrêter	Mode de fonctionnement manuel
L'utilisateur change la vitesse	La vitesse de changement des images varie en fonction du choix de l'utilisateur
Clic sur vider le lecteur	L'utilisateur choisit de décharger le diaporama et le lecteur se vide

Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>
Charger le diaporama	Chargement manuel du diaporama par l'utilisateur
Mode par défaut sélectionné	Le mode par défaut est le mode manuel
mode de fonctionnement manuel	Le mode passe à manuel
mode de fonctionnement automatique	Le mode passe à auto
passer à l'image précédente	L'image précédente s'affiche
passer à l'image suivante	L'image suivante s'affiche
quitter l'application	L'application se ferme
Vider le lecteur	Le lecteur devient vide, il faut à nouveau charger le diaporama

2.2 Table T_EtatsEvenementsActions (v4)

<i>Élément graphique pregnant en charge cet événement</i>	X	Fichier >> Paramétrer >> Charger le diaporama		Bouton <	Bouton >	Fichier >> Quitter	Bouton Lancer	Bouton arrêter	Fichier >> Paramétrer >> Vitesse	Fichier >> Paramétrer >> Enlever diaporama
<i>Événement nomEtat</i>	Ouverture de l'application	L'utilisateur charger le diaporama	Mode manuel par défaut	Clic sur précédent	Clic sur suivant	Clic sur quitter	Clic sur lancer	Clic sur arrêter	L'utilisateur changer la vitesse	Clic sur vider le lecteur
Diaporama non chargé		Chargement du diaporama								Retour à non chargé
Diaporama chargé										
Mode manuel			Mode par défaut				Passage au mode auto			
Mode auto				Retour au mode manuel	Retour au mode manuel	Quitter l'application		Retour au mode manuel		
Image précédente				Passer à l'image précédente						
Image suivante					Passer à l'image suivante					
Quitter						Quitter l'appl				

						ication				
Image suivante									Change ment de la vitesse de défile ment des images	

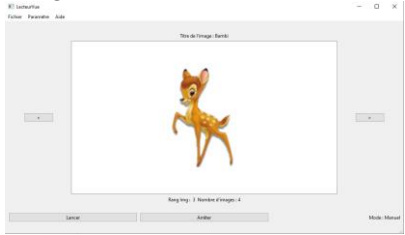

3. Implémentation et tests

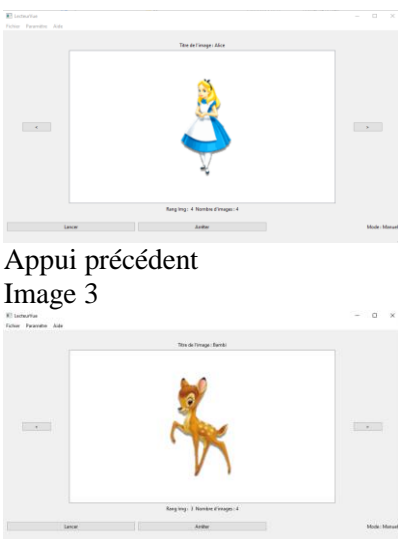
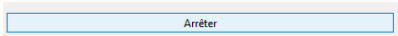
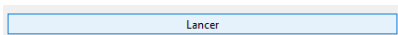
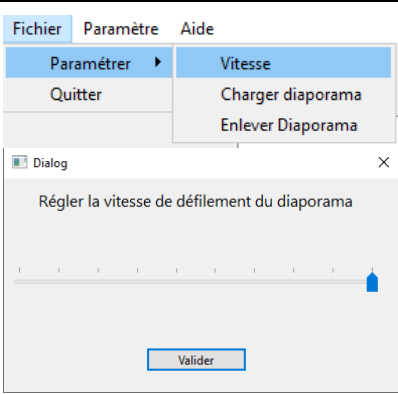
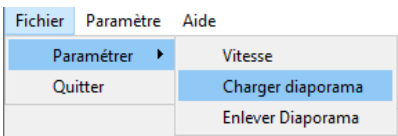
3.1 Implémentation (v4)

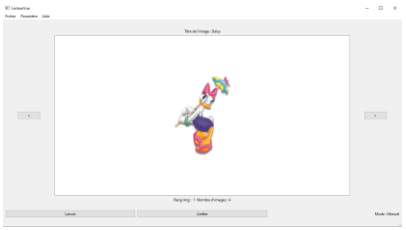
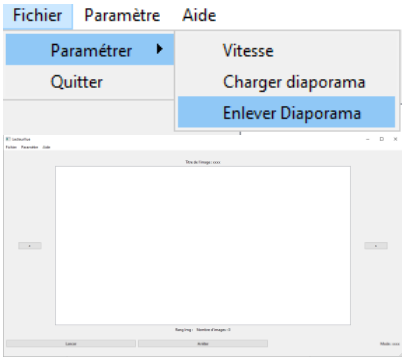
Liste et rôle des fichiers de cette version :

lecteurVue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas ROLE IDEM A V2
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 ROLE IDEM A V2
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image ROLE IDEM A V2
image.cpp	Corps de la classe Image
main.cpp	ROLE IDEM A V2

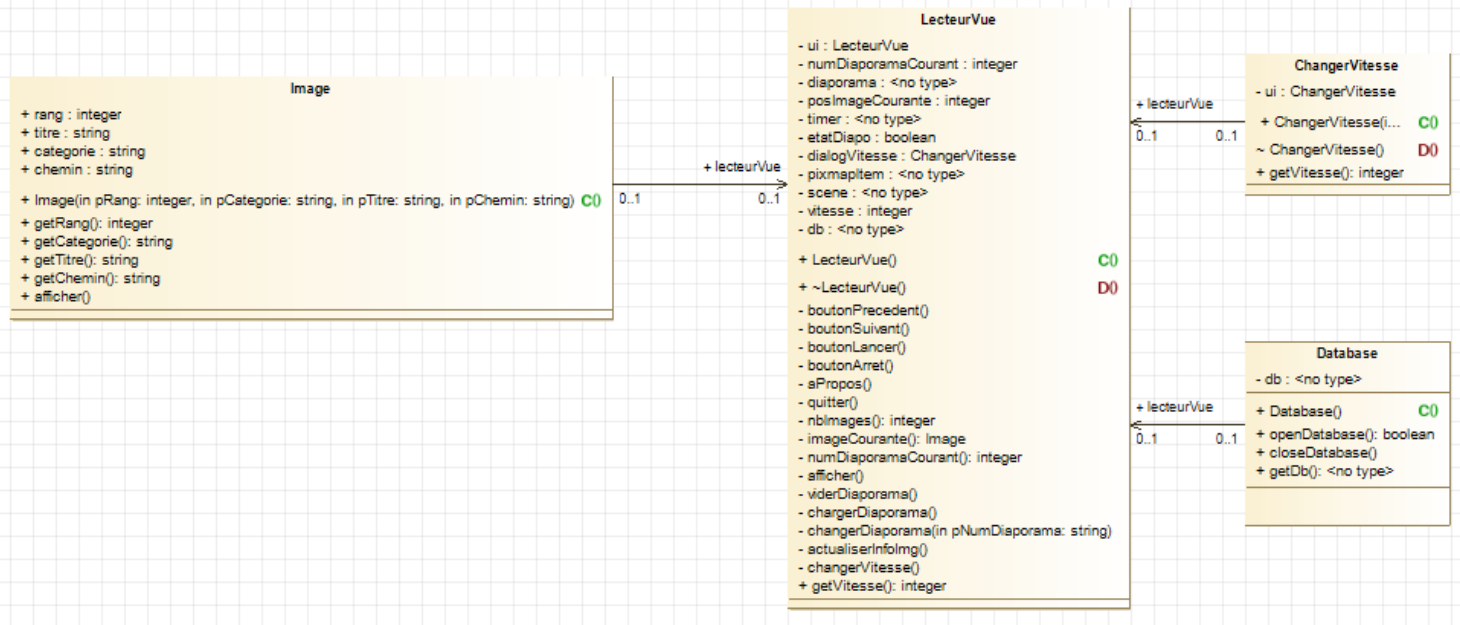
3.2 Tests (v4)

Action réalisée	Réponse attendue	Réponse obtenue	Preuve
clic du bouton suivant	Image suivante	Image suivante	<p>Image 3</p>  <p>Appui suivant Image 4</p> 
clic du bouton précédent	Image précédente	Image précédente	Image 4

			 <p>Appui précédent Image 3</p>
clic du bouton arrêter	Passe au mode manuel	Passe au mode manuel	<p>Etat Diapo :</p> <p>Mode : Automatique</p> <p>Clic sur bouton arrêter :</p>  <p>Etat Diapo :</p> <p>Mode : Manuel</p>
clic du bouton lancer	Passe au mode automatique	Passe au mode automatique	<p>Etat Diapo :</p> <p>Mode : Manuel</p> <p>Clic sur bouton lancer:</p>  <p>Etat Diapo :</p> <p>Mode : Automatique</p>
Changement de vitesse (Fichier>>Paramétrer>>Vitesse)	Ouverture du dialogue vitesse	Ouverture du dialogue vitesse	
Charger le diaporama (Fichier>>Paramétrer>>Charger diaporama)	Diaporama se charge	Diaporama se charge	

			
Vider le diaporama (Fichier>>Paramétrer>> Vider diaporama	Diaporama se vide	Diaporama se vide	

1. Diagramme de classes (UML)



2. Comportement de l'application

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

IDEM A V4

2.2 Dictionnaire des états, événements et Actions (v5)

IDEM A V4

3. Implémentation et tests

3.1 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 ROLE IDEM A V2
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 ROLE IDEM A V2
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image ROLE IDEM A V2
image.cpp	Corps de la classe Image
main.cpp	ROLE IDEM A V2
changerVitesse.h	Spécification de la classe ChangerVitesse ChangerVitesse est la boîte de dialogue qui permet à l'utilisateur de changer la vitesse de défilement du diaporama lors du mode auto

changerVitesse.cpp	Corps de la classe ChangerVitesse
chnagervitesse.ui	Fichier du dessin de la boite de dialogue réalisé par QtDesigner
database.h	Spécification de la classe Database Database est la classe qui permet de gérer la connexion à la base de données
database.cpp	Corps de la classe Database

3.2 Tests (v5)

IDEM A V4

Bilan

Le projet complet se trouve sur le github suivant :
<https://github.com/KyoooCode/S201DevApp/>

Nous avons travaillé durant les heures en autonomie, mais aussi chez nous soit environ une quinzaine d'heures au total.

Nous avons pu lors de cette SAE perfectionner nos connaissances en QT. Nous n'avons pas réellement appris de nouvelles choses, mais la manipulation du logiciel nous a permis d'apprendre à beaucoup mieux utiliser l'IDE ainsi que c++.

Nous pensons que la difficulté majeure de cette SAE n'est pas dans le code, mais dans le rapport. En effet, le rapport est très conséquent et difficile à remplir, il est parfois compliqué de comprendre ce qui est attendu dans le rapport.

Points positifs	Points négatifs
<ul style="list-style-type: none">- Progrès effectués que cela soit en QT ou en c++- Suivi par un professeur assez régulier- Sujet intéressant- Apprentissage de github MAIS ----->	<ul style="list-style-type: none">- Rapport beaucoup trop long et difficile- Pas suffisamment d'heures pour terminer <p>Deux cours de git uniquement, donc difficile à mettre en oeuvre au début donc perte de beaucoup de temps</p>