

Saé 2.01 – Développement d'une application

Lecteur de diaporamas – Dossier d'Analyse et conception

dépôt Github : https://github.com/DEVSAEIHM/sae_developpement_application_ihm

Département Informatique
Damageux Lucas
Dirchaoui El Mahdi
Laborde Romain



SOMMAIRE

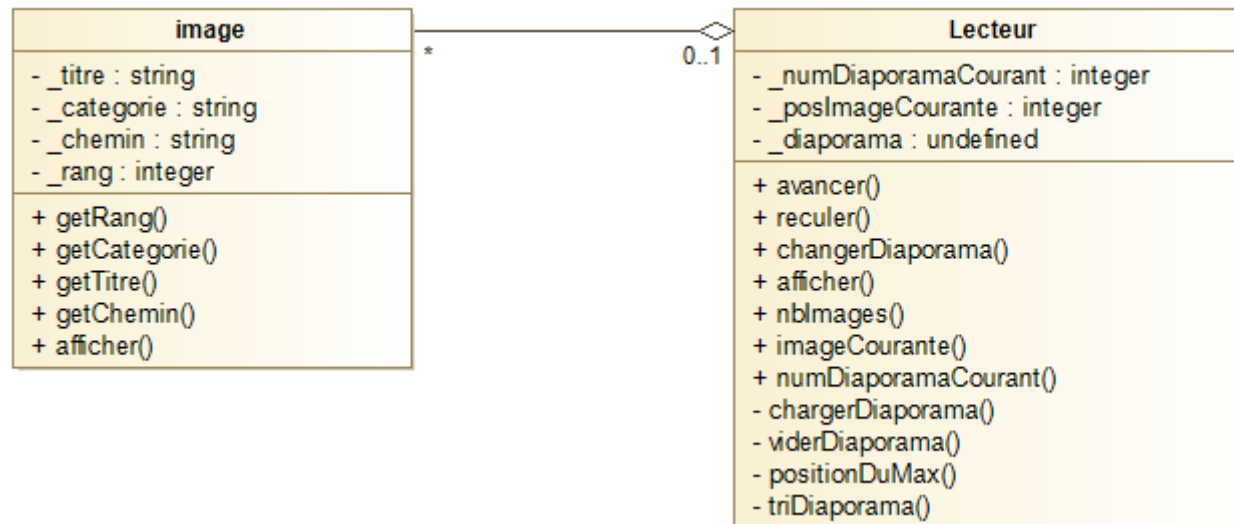
1. COMPLÉMENTS DE SPÉCIFICATIONS EXTERNES.....	3
1.1 Scénarios.....	3
2. Version v0 – Version console seule.....	9
2.1 Implémentation et tests (v0).....	9
2.1.1 Implémentation.....	9
2.1.2 Tests.....	9
3. VERSION 1 - PROJET GRAPHIQUE.....	12
3.1 Éléments d’interfaces.....	12
3.2 Implémentation et tests (v1).....	13
3.2.1 Implémentation.....	13
3.2.2 Tests (v1).....	14
4. VERSION V2 –.....	15
4.1 Diagramme de classes (UML).....	15
4.2 Comportement de l’application.....	16
4.2.1 Diagramme états-transitions-actions du lecteur de diaporamas (v2).....	16
4.2.2 Dictionnaire des états, événements et Actions (v2).....	17
4.2.3 Table T_EtatsEvenementsActions (v2).....	18
4.3 Implémentation et tests.....	19
4.3.1 Implémentation.....	19
4.3.2 Tests (v2).....	20
5. VERSION V3 –.....	25
5.1 Diagramme de classes (UML).....	25
5.2 Comportement de l’application.....	26
5.2.1 Diagramme états-transitions-actions du lecteur de diaporamas (v3).....	26
5.2.2 Dictionnaire des états, événements et Actions (v3).....	28
5.2.3 Table T_EtatsEvenementsActions (v3).....	31
5.3 Implémentation et tests.....	32

5.3.1 Implémentation.....	32
5.3.2 Tests (v3).....	33
6. VERSION V4 –.....	36
6.1 Diagramme de classes (UML).....	36
6.2 Comportement de l'application.....	38
6.2.1 Diagramme états-transitions-actions du lecteur de diaporamas (v4).....	38
6.2.2 Dictionnaire des états, événements et Actions (v4).....	40
6.2.3 Table T_EtatsEvenementsActions (v4).....	43
6.3 Implémentation et tests.....	45
6.3.1 Implémentation.....	45
6.3.2 Tests (v4).....	46
7. VERSION V5 –.....	49
7.1 Diagramme de classes (UML).....	49
7.2 Comportement de l'application.....	51
7.2.1 Diagramme états-transitions-actions du lecteur de diaporamas (v5).....	51
7.2.2 Dictionnaire des états, événements et Actions (v5).....	53
7.2.3 Table T_EtatsEvenementsActions (v5).....	57
7.3 Implémentation et tests.....	57
7.3.1 Implémentation.....	57
7.3.2 Tests (v5).....	58
8. VERSION V6 –.....	59
8.1 Diagramme de classes (UML).....	59
8.2 Comportement de l'application.....	61
8.2.1 Diagramme états-transitions-actions du lecteur de diaporamas (v5).....	61
8.2.2 Dictionnaire des états, événements et Actions (v5).....	63
8.2.3 Table T_EtatsEvenementsActions (v5).....	67
8.3 Implémentation et tests.....	67
8.3.1 Implémentation.....	67
8.3.2 Tests (v5).....	68
9. BILAN.....	69

1. COMPLÉMENTS DE SPÉCIFICATIONS EXTERNES

1.1 Scénarios

Cas d'utilisation	lire un diaporama en mode manuel	
Acteur primaire	utilisateur	
Système	Lecteur Diaporama	
Acteur secondaire		
Niveau	OBJECTIF UTILISATEUR	
Préconditions	Avoir un diaporama à vouloir lancer.	
Opérations	Acteur	Système
1	L'utilisateur ouvre l'application	
2	L'utilisateur clique sur charger diaporama	
3		Le système charge le diaporama de l'utilisateur et le mode sélectionné est le mode manuel.
4	L'utilisateur parcourt son diaporama à l'aide des boutons "suivant" et "precedent"	
5		Le système défile les pages comme l'utilisateur le souhaite
6	L'utilisateur clique sur enlever diaporama	
7		Le système enlève le diaporama
8	L'utilisateur clique sur quitter	
9		Le système ferme l'application.
10		
11		
Extension		
5.A	L'utilisateur veut passer le lecteur en mode automatique	
5.A.1	L'utilisateur clique sur "lancer diaporama"	
5.A.2		Le système passe en mode auto
5.A.3		Le système rend actif le bouton "arrêter diaporama"
5.A.4		Le système parcourt une par une toutes les images du diaporama dans l'ordre
5.A.5	Retour à l'étape 7	
Extension		
5.A.0	L'utilisateur veut changer la vitesse de défilement du lecteur en mode automatique	
Préconditions	Le lecteur doit être en mode automatique	
3.A.1	L'utilisateur clique sur le bouton "Vitesse"	
3.A.2		Le système ouvre une boîte de dialogue permettant de modifier la vitesse de défilement des images
3.A.3	L'utilisateur sélectionne la vitesse de défilement souhaitée et valide son choix	
3.A.4		Le système parcourt une par une toutes les images du diaporama dans l'ordre avec la vitesse souhaitée par l'utilisateur
3.A.5	Retour à l'étape 7	



_diaporama est de type [Diaporama](#), nous ne l'avons pas mis sur modelio car le type Diaporama n'est pas disponible.

(a) Dictionnaire des éléments pour chaque classe

Classe Lecteur			
Nom attribut	Signification	Type	Exemple
_numDiaporamaCourant	numéro du diaporama courant, par défaut 0	unsigned int	1
_diaporama	pointeur vers les images du diaporama	Diaporama	
_posImageCourante	position dans le diaporama de l'image courante, démarre à 0 quand diaporama non vide	unsigned int	0

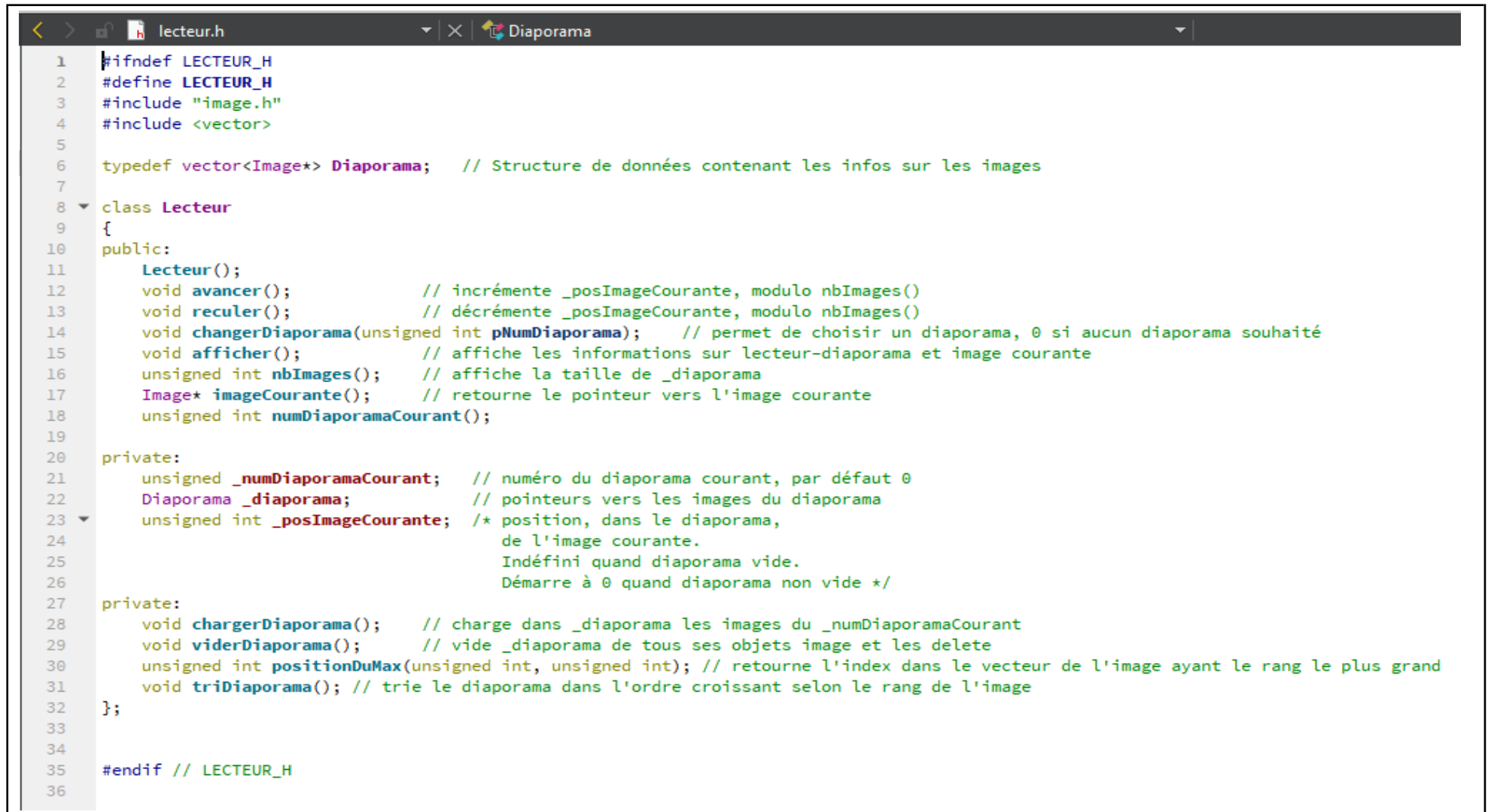
Tableau 1 : Dictionnaire des éléments - Classe Lecteur

Classe Image			
Nom attribut	Signification	Type	Exemple
_rang	rang de l'image au sein du diaporama	unsigned int	2
_titre	intitulé de l'image	string	"img1"
_categorie	catégorie de l'image (animal, personne, objet)	string	"animal"
_chemin	chemin complet vers le dossier où se trouve l'image	string	"./doc/img.png"

Tableau 2 : Dictionnaire des éléments - Classe Image

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

Exemple (classe lecteur de la version Console) :



```
1  #ifndef LECTEUR_H
2  #define LECTEUR_H
3  #include "image.h"
4  #include <vector>
5
6  typedef vector<Image*> Diaporama;  // Structure de données contenant les infos sur les images
7
8  class Lecteur
9  {
10 public:
11     Lecteur();
12     void avancer();           // incrémente _posImageCourante, modulo nbImages()
13     void reculer();           // décrémente _posImageCourante, modulo nbImages()
14     void changerDiaporama(unsigned int pNumDiaporama); // permet de choisir un diaporama, 0 si aucun diaporama souhaité
15     void afficher();          // affiche les informations sur lecteur-diaporama et image courante
16     unsigned int nbImages();   // affiche la taille de _diaporama
17     Image* imageCourante();    // retourne le pointeur vers l'image courante
18     unsigned int numDiaporamaCourant();
19
20 private:
21     unsigned _numDiaporamaCourant; // numéro du diaporama courant, par défaut 0
22     Diaporama _diaporama;          // pointeurs vers les images du diaporama
23     unsigned int _posImageCourante; /* position, dans le diaporama,
24                                     de l'image courante.
25                                     Indéfini quand diaporama vide.
26                                     Démarre à 0 quand diaporama non vide */
27
28 private:
29     void chargerDiaporama(); // charge dans _diaporama les images du _numDiaporamaCourant
30     void viderDiaporama();   // vide _diaporama de tous ses objets image et les delete
31     unsigned int positionDuMax(unsigned int, unsigned int); // retourne l'index dans le vecteur de l'image ayant le rang le plus grand
32     void triDiaporama();     // trie le diaporama dans l'ordre croissant selon le rang de l'image
33
34 };
35
36 #endif // LECTEUR_H
```

Figure 1 : Schéma de classes = Classe Lecteur

```
< > image.h using namespace std
1  #ifndef IMAGE_H
2  #define IMAGE_H
3  #include <iostream>
4  using namespace std;
5
6  class Image
7  {
8  public:
9      Image(unsigned int pRang=0,
10             string pCategorie="", string pTitre="", string pChemin = "");
11      unsigned int getRang();
12      string getCategorie();
13      string getTitre();
14      string getChemin();
15      void afficher();           // affiche tous les champs de l'image
16
17  private:
18      unsigned int _rang;        /* rang de l'image au sein du diaporama
19                                  auquel l'image est associée */
20      string _titre;             // intitulé de l'image
21      string _categorie;         // catégorie de l'image (personne, animal, objet)
22      string _chemin;           // chemin complet vers le dossier où se trouve l'image
23  };
24
25  #endif // IMAGE_H
26
```

Figure 2 : Schéma de classes = Classe Image

2. Version v0 – Version console seule

2.1 ***Implémentation et tests (v0)***

2.1.1 ***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

2.1.2 ***Tests***

Test avec le programme fourni main.cpp

lecteurDiaporama [main]

lecteurDiaporama.pro

Headers

image.h

lecteur.h

Sources

image.cpp

lecteur.cpp

main.cpp

13

void L Lecteur::reculer()

14

{

15

_posImageCourante = (_posImageCourante -1) % nbImages();

16

}

17

18

void L Lecteur::changerDiaporama(unsigned int pNumDiaporama)

19

{

20

// s'il y a un diaporama courant, le vider, puis charger le nouveau Diaporama

21

if (numDiaporamaCourant() > 0)

22

{

23

viderDiaporama();

Sortie de l'application

1-testSelectBDD

lecteurvue

lecteurDiaporama

10:20:54: Starting F:\courBut\semestre2\s201\v0_version_console_pourEtudiants\build-lecteurDiaporama-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurDiaporama.exe...

Lecteur vide

Diaporama num. 1 selectionne.

4 images chargees dans le diaporama

Diaporama num. 1

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

Test avancer() : 4 fois

avancer() :

Diaporama num. 1

image(rang:2, titre:Cendrillon, categorie:personne, chemin:C:\cartesDisney\carteDisney4.gif)

avancer() :

Diaporama num. 1

image(rang:3, titre:Blanche Neige, categorie:personne, chemin:C:\cartesDisney\carteDisney2.gif)

avancer() :

Diaporama num. 1

image(rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)

avancer() :

Diaporama num. 1

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

Open Documents

lecteur.cpp

lecteurDiaporama [main]

- lecteurDiaporama.pro
- Headers
 - image.h
 - lecteur.h
- Sources
 - image.cpp
 - lecteur.cpp
 - main.cpp

```

13 void Lecteur::reculer()
14 {
15     _posImageCourante = (_posImageCourante - 1) % nbImages();
16 }
17
18 void Lecteur::changerDiaporama(unsigned int pNumDiaporama)
19 {
20     // s'il y a un diaporama courant, le vider, puis charger le nouveau Diaporama
21     if (numDiaporamaCourant() > 0)
22     {
23         viderDiaporama();

```

Sortie de l'application

1-testSelectBDD x lecteurvue x lecteurDiaporama x

```

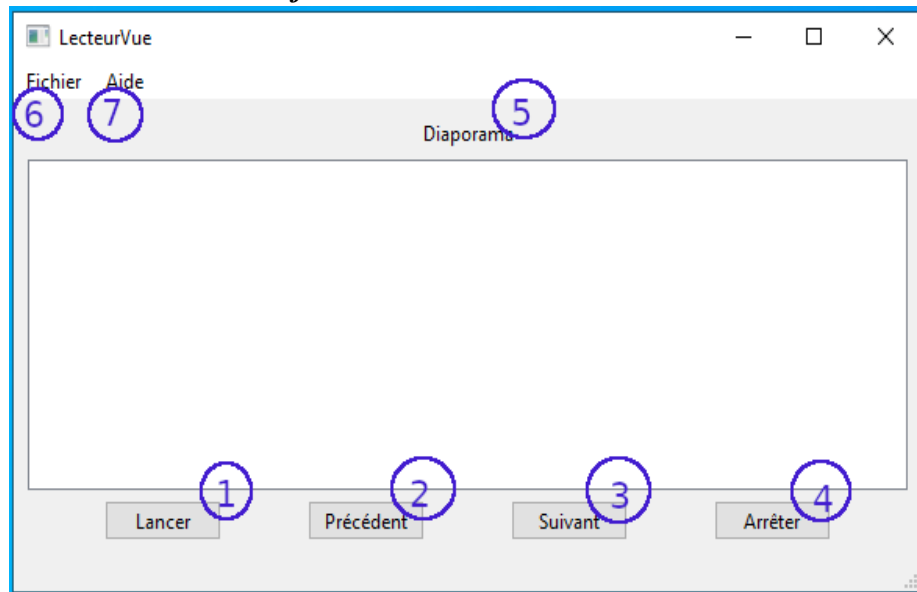
Test reculer() : 5 fois
reculer() :
Diaporama num. 1
image( rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)
reculer() :
Diaporama num. 1
image( rang:3, titre:Blanche Neige, categorie:personne, chemin:C:
\cartesDisney\carteDisney2.gif)
reculer() :
Diaporama num. 1
image( rang:2, titre:Cendrillon, categorie:personne, chemin:C:
\cartesDisney\carteDisney4.gif)
reculer() :
Diaporama num. 1
image( rang:1, titre:Grincheux, categorie:personne, chemin:C:
\cartesDisney\carteDisney1.gif)
reculer() :
Diaporama num. 1
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
10:20:54: F:\courBut\semestre2\s201\v0_version_console_pourEtudiants\build-
lecteurDiaporama-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurDiaporama.exe exited with
code 0

```

3. VERSION 1 - PROJET GRAPHIQUE

3.1 Éléments d'interfaces



LecteurVue	QMainWindow
centralwidget	QWidget
gridLayout	QGridLayout
horizontalLayout_7	QHBoxLayout
btnArreter	QPushButton
btnLancer	QPushButton
btnPrecedent	QPushButton
btnSuivant	QPushButton
horizontalSpacer	Spacer
horizontalSpacer_2	Spacer
horizontalSpacer_3	Spacer
horizontalSpacer_4	Spacer
horizontalSpacer_5	Spacer
horizontalLayout_8	QHBoxLayout
horizontalSpacer_6	Spacer
horizontalSpacer_7	Spacer
intitule	QLabel
imgVue	QGraphicsView
menubar	QMenuBar
menu_Aide	QMenu
menu_Fichier	QMenu
statusbar	QStatusBar

	Intitulé	Nom objet	Description
1	Bouton Lancer	btnLancer	Permet de lancer la lecture du diaporama en mode "Auto"
2	Bouton Précédent	btnPrecedent	Permet d'afficher l'image précédente, mais aussi lorsque le lecteur est en mode "Auto" de repasser en mode "Manuel".
3	Bouton Suivant	btnSuivant	Permet d'afficher l'image suivante, mais aussi lorsque le lecteur est en mode "Auto" de repasser en mode

			“Manuel”.
4	Bouton Arrêter	btnArreter	Permet d’arrêter la lecture en mode “Auto” et de passer en mode “Manuel”.
5	Label Diaporama	intitule	Permet d’afficher le numéro du diaporama courant ainsi que le nom de l’image et d’autres informations.
6	Menu fichier	menu_Fichier	Permet de dérouler un sous-menu avec plusieurs actions possibles comme quitter l’application.
7	Menu Aide	menu_Aide	Permet de dérouler un sous-menu contenant l’action d’afficher la fenêtre de dialogue “A propos de...”

3.2 Implémentation et tests (v1)

3.2.1 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	Teste les méthodes de la classe Lecteur

Remarques sur l’implémentation :

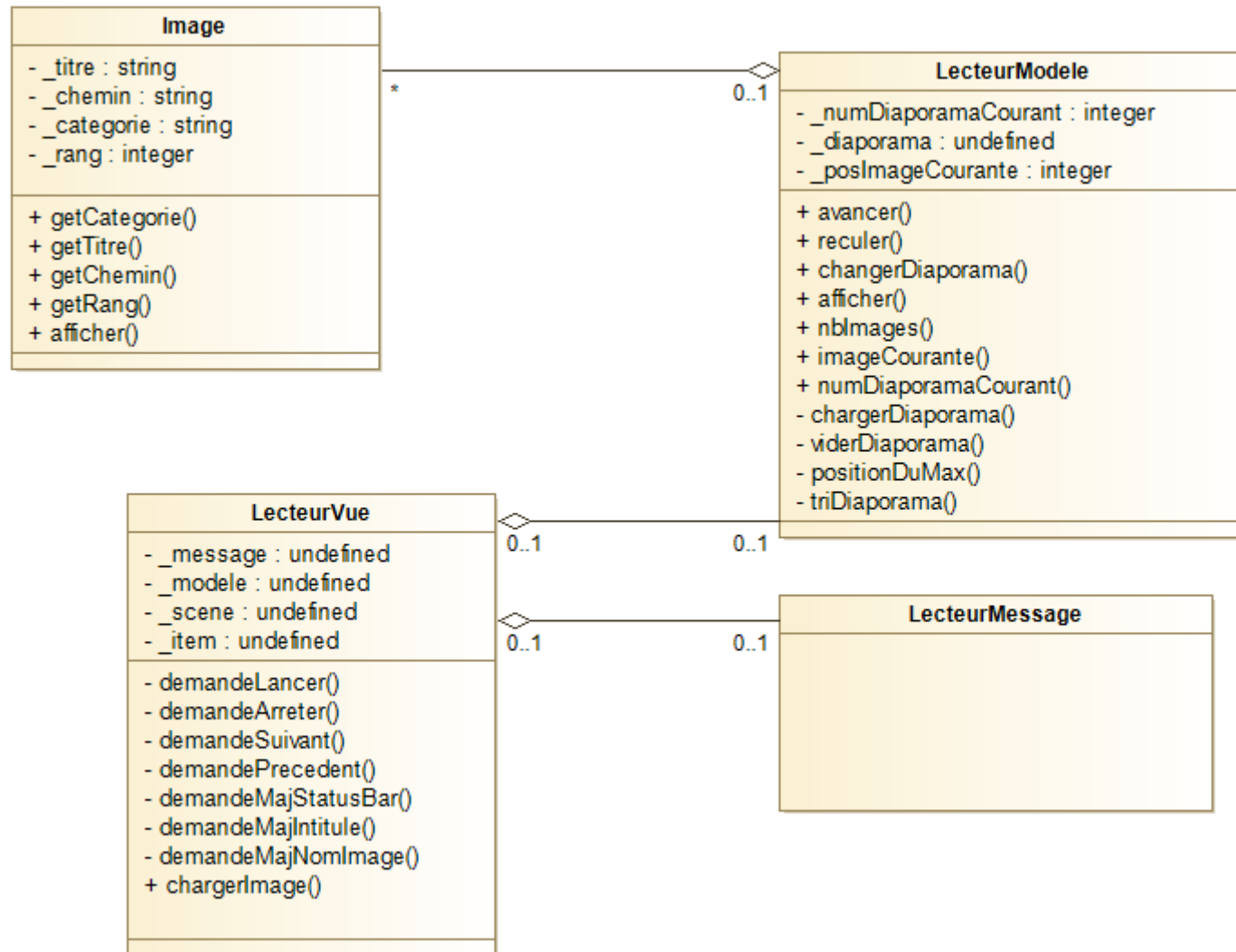
Dans cette première version de la LecteurVue, nous avons connecté les signaux de chaque bouton de la fenêtre graphique à des slots respectifs. Nous avons fait le choix de cette implémentation, car chacun de ces boutons va déclencher des actions différentes.

3.2.2 Tests (v1)

Opération testé	Description	Comportement attendu	Comportement observé
Passage à l'image suivante	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action de passer à l'image suivante a été demandée.	Affichage dans la console du message "action aller à l'image suivante".	10:11:46: Starting F: \courBut\semestre2\s201\v1_lecteurvue\build-lecteurvue-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... action aller à l'image suivante <i>impression écran de la console de debug</i>
Passage à l'image précédente	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action de passer à l'image précédente a été demandée.	Affichage dans la console du message "action aller à l'image précédente".	10:10:53: Starting F: \courBut\semestre2\s201\v1_lecteurvue\build-lecteurvue-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... action aller à l'image précédente <i>impression écran de la console de debug</i>
Lancement lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action de lancer la lecture du diaporama a été demandée.	Affichage dans la console du message "action lancer le diaporama".	10:09:48: Starting F: \courBut\semestre2\s201\v1_lecteurvue\build-lecteurvue-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... action lancer le diaporama <i>impression écran de la console de debug</i>
Arrêt lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action d'arrêter la lecture du diaporama a été demandée.	Affichage dans la console du message "action arreter le diaporama".	10:12:20: Starting F: \courBut\semestre2\s201\v1_lecteurvue\build-lecteurvue-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... action arreter le diaporama <i>impression écran de la console de debug</i>

4. VERSION V2 –

4.1 Diagramme de classes (UML)



`_diaporama` est de type `Diaporama`, nous ne l'avons pas mis sur modelio car le type `Diaporama` n'est pas disponible. `_message` est de type `LecteurMessage`, `_modele` est de type `LecteurModele`, `_scene` est de type `QGraphicsScene` et `_item` est de type `QGraphicsPixmapItem`.

4.2 Comportement de l'application

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

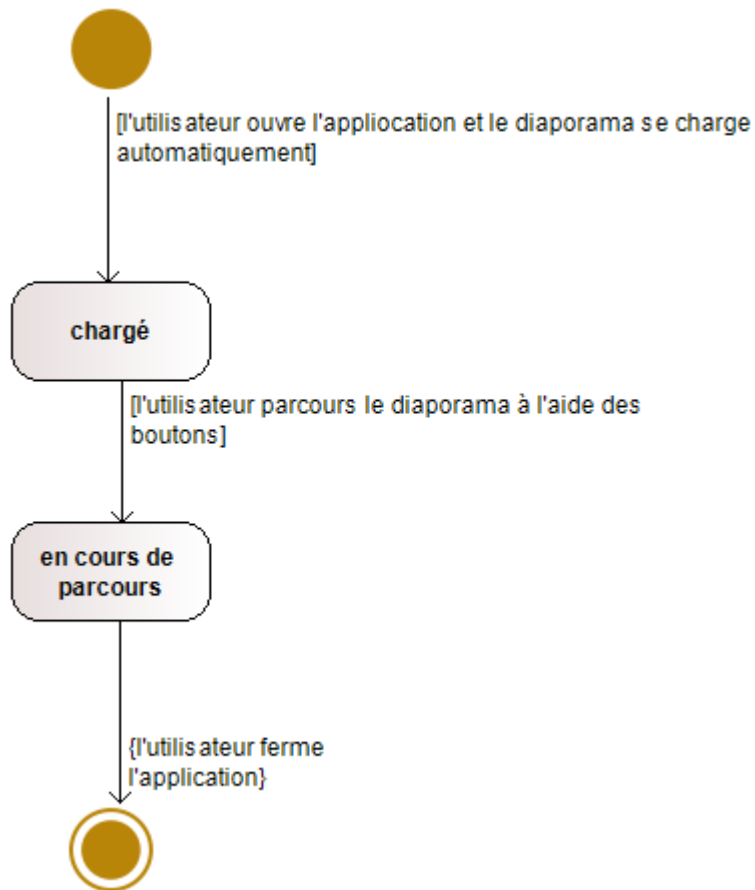


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v2

4.2.2 Dictionnaire des états, événements et Actions (v2)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
chargé	un diaporama est chargé
en cours de parcours	L'utilisateur est en train de parcourir le diaporama en mode manuel (parcours des images)
Fermé	l'utilisateur ferme l'application

Tableau 3 : États du lecteur de diaporamas – v2

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomEvénement</i>	<i>Signification</i>
L'utilisateur ouvre l'application	L'utilisateur ouvre l'application
L'utilisateur parcourt le diaporama à l'aide des boutons	L'utilisateur passe les images à l'aide des boutons
l'utilisateur ferme l'application	l'utilisateur ferme l'application

Tableau 4 : Événements faisant changer le diaporama d'état – v2

Description des actions réalisées lors de la traversée des transitions 1er partie

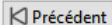
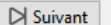
<i>nomAction</i>	<i>Signification</i>
ouverture et chargement du diaporama	ouverture et chargement du diaporama
passage au diaporama suivant	le lecteur diaporama passe à l'image suivante
passage au diaporama précédent	le lecteur diaporama retourne à l'image précédente
l'application se ferme	l'application se ferme

Tableau 5 : Actions à réaliser lors des changements d'état – lecteur de diaporamas v2

4.2.3 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> <input type="checkbox"/>		 	
<i>Événement</i> <input type="checkbox"/> <i>nomEtat</i>	l'utilisateur ouvre l'application	l'utilisateur parcourt le diaporama en mode manuel	l'utilisateur ferme l'application

<i>chargé</i>	ouverture et chargement du diaporama		
<i>en cours de parcours</i>		-passage à l'image suivant -passage à l'image précédent	
Fermé			l'application se ferme

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.

4.3 **Implémentation et tests**

4.3.1 **Implémentation**

Liste et rôle des fichiers de cette version :

lecteurvue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de
--------------	---

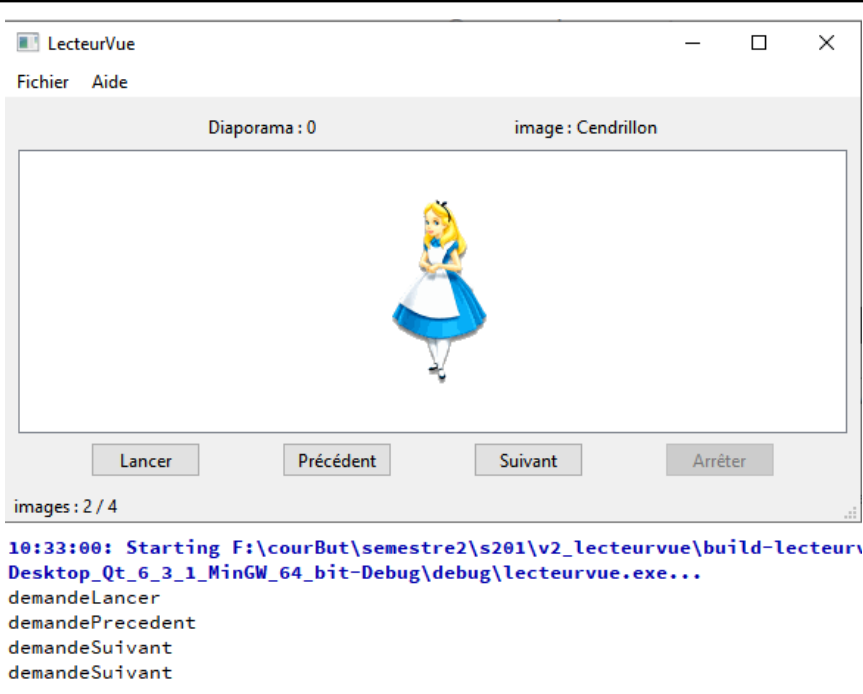
	diaporamas Concerne la gestion graphique de l'application.
lecteurvue.cpp	Corps de la classe LecteurVue.
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner
lecteurmodele.h	Spécification de la classe LecteurModele. Est centré sur informations et les actions métier et déconnecté du dialogue utilisateur.
lecteurmodele.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image Permet d'accéder aux diverses informations de l'image, comme le nom, son chemin, sa catégorie, via ses méthodes.
image.cpp	Corps de la classe Image
main.cpp	Permet de créer l'application et la fenêtre principale LecteurVue.

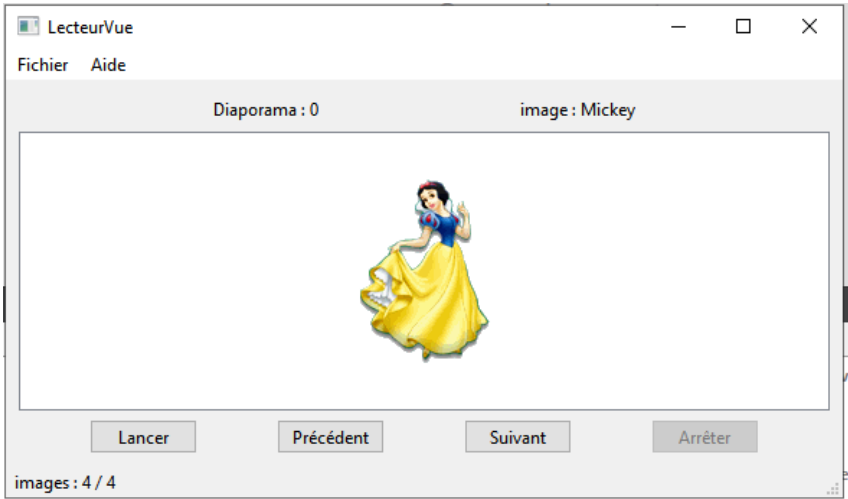
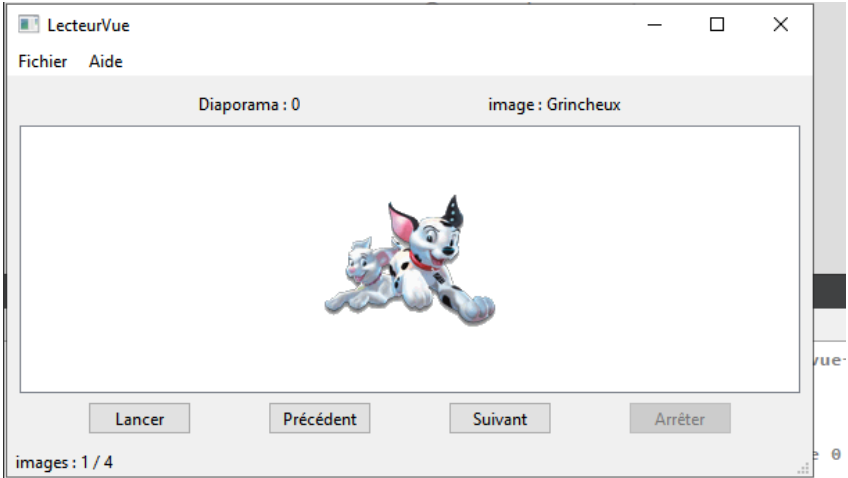
Remarques sur l'implémentation :

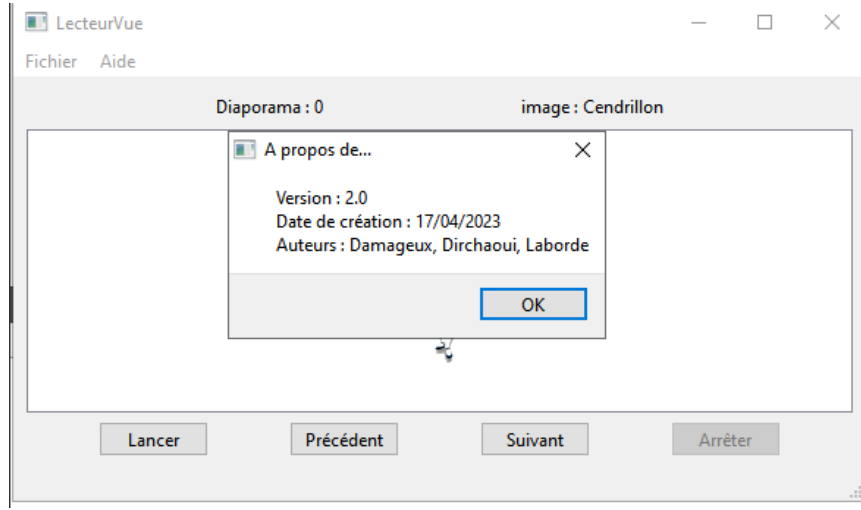
Commenter brièvement les choix importants d'implémentation réalisés, comme par exemple, les signals/slots

4.3.2 Tests (v2)

Opération testé	Description	Comportement attendu	Comportement observé
-----------------	-------------	----------------------	----------------------

Passage à l'image suivante	Dans cette version, cette opération change l'image courante affichée et affiche l'image suivante.	Affichage de l'image suivante.	
----------------------------	---	--------------------------------	---

Passage à l'image précédente	Dans cette version, cette opération change l'image courante affichée et affiche l'image précédente.	Affichage de l'image précédente.	 <pre> 10:33:00: Starting F:\courBut\semestre2\s201\v2_lecteurvue\build-lecteurvue-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... demandeLancer demandePrecedent </pre>
Lancement lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action de lancer la lecture du diaporama a été demandée.	Affichage dans la console du message "demandeLancer".	 <pre> 10:33:00: Starting F:\courBut\semestre2\s201\v2_lecteurvue\build-lecteurvue-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... demandeLancer </pre>

Arrêt lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action d'arrêter la lecture du diaporama a été demandée.	Pour cette version le bouton arrêter est désactivé au démarrage de l'application, nous ne le testerons pas.	<i>Pour cette version le bouton arrêter est désactivé au démarrage de l'application, nous ne le testerons pas.</i>
Quitter l'application	L'appui dans le menu fichier de l'action "Quitter" arrête l'application	Arrêt de l'application, la fenêtre principale se ferme.	10:36:21: F:\courBut\semestre2\s201\v2_lecteurvue\build-lecteurvue-Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe exited with code 0
Afficher informations application	L'appui dans le menu aide de l'action "A propos de..." ouvre une boîte de message donnant les informations sur l'application tel que : sa version ; date de création ; les auteurs.	Affichage d'une fenêtre de dialogue contenant les informations sur l'application.	

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 programme
 - Le comportement fonctionnel de l'application

5. VERSION V3 –

5.1 Diagramme de classes (UML)



`_diaporama` est de type `Diaporama`, nous ne l'avons pas mis sur modelio car le type `Diaporama` n'est pas disponible. `_message` est de type `LecteurMessage`, `_modele` est de type `LecteurModele`, `_scene` est de type `QGraphicsScene` et `_item` est de type `QGraphicsPixmapItem`.
`_timer` est de type `QTimer`. `_mode` est de type `QString`.

5.2 Comportement de l'application

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

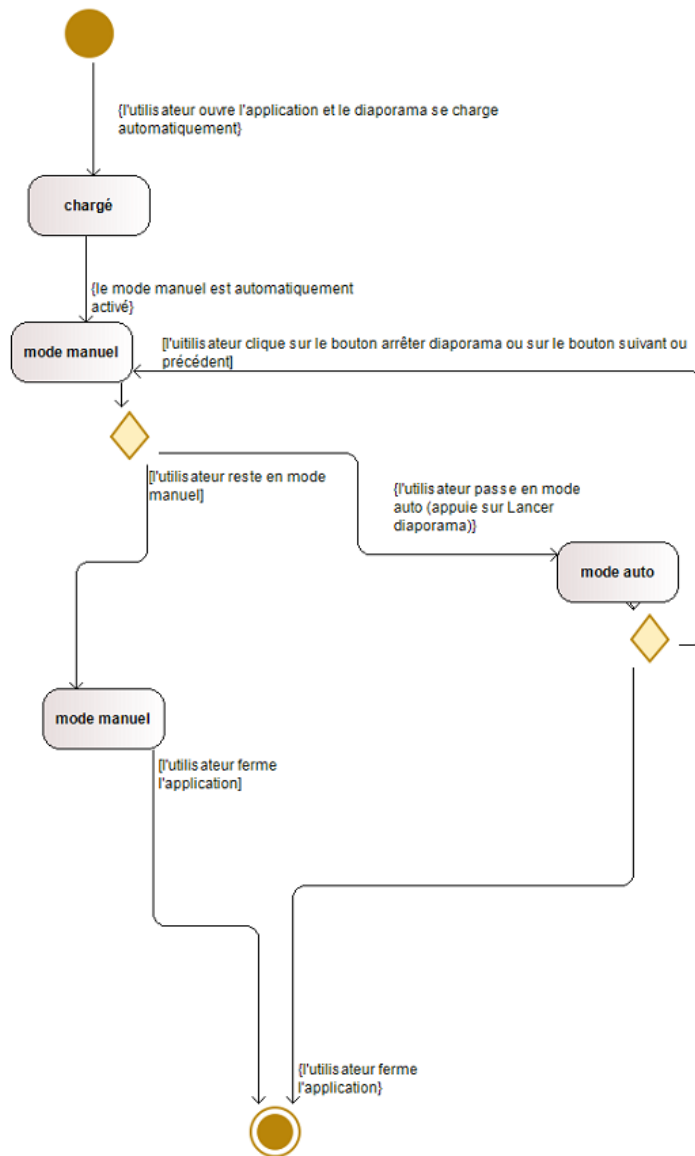


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v3

5.2.2 Dictionnaire des états, événements et Actions (v3)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
chargé	un diaporama est chargé
Mode manuel	Le diaporama se parcourt grâce aux boutons et c'est l'utilisateur qui doit les activer
Mode auto	Le diaporama se parcourt automatiquement avec la vitesse par défaut
Fermé	l'utilisateur ferme diaporama

Tableau 2 : États du lecteur de diaporamas – v3

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomÉvénement</i>	<i>Signification</i>
L'utilisateur ouvre l'application	L'utilisateur ouvre l'application
L'utilisateur parcourt le diaporama	L'utilisateur passe les images à l'aide des boutons

L'utilisateur passe en mode auto	L'utilisateur passe en mode auto avec la vitesse de défilement par défaut
L'utilisateur clique sur le bouton arrêter diaporama ou sur le bouton suivant ou précédent	L'utilisateur passe en mode manuel
L'utilisateur ferme l'application	L'utilisateur ferme l'application

Tableau 3 : Événements faisant changer le diaporama d'état – v3

Description des actions réalisées lors de la traversée des transitions 1er partie

<i>nomAction</i>	<i>Signification</i>
l'application s'ouvre	l'application est ouverte
le diaporama se charge	le diaporama est chargé dans le lecteur diaporama
passage en mode manuel	le mode manuel est activé, lecteur attend les interactions de l'utilisateur
passage à l'image suivante	le lecteur diaporama passe à l'image suivante
passage à l'image précédente	le lecteur diaporama retourne à l'image précédente
l'arrêt du diaporama	le lecteur diaporama arrête la lecture du diaporama

Tableau 4 : Actions à réaliser lors des changements d'état – lecteur de diaporamas v3

Description des actions réalisées lors de la traversée des transitions 2ème partie

<i>nomAction</i>	<i>Signification</i>
l'application s'ouvre et charge un diaporama	l'application s'ouvre et charge un diaporama
lecteur passe au mode auto	lecteur diaporama passe en mode automatique
passage lecture auto normal	une lecture en mode auto du diaporama avec une vitesse normale prédéfinie
diaporama s'arrête	le lecteur diaporama s'arrête
diaporama se ferme	l'application se ferme

5.2.3 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

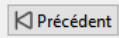
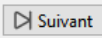
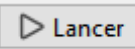
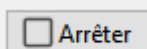
<i>Élément graphique pregnant en charge cet événement</i> <input type="checkbox"/>		 			
<i>Événement</i> <input type="checkbox"/> <i>nomEtat</i>	L'utilisateur ouvre l'application	l'utilisateur ne passe pas en mode auto	l'utilisateur passe en mode auto	L'utilisateur enlève le diaporama	l'utilisateur ferme l'application
<i>chargé</i>	l'application s'ouvre et charge un diaporama				
<i>mode manuel</i>		-passage à l'image suivante -passage à l'image précédente			
<i>mode auto</i>		passage en mode manuel	passage en mode auto	l'arrêt du diaporama	
Fermé					l'application se ferme

Tableau 5 : Matrice d'états-transitions du lecteur de diaporamas – v3

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

5.3 Implémentation et tests

5.3.1 Implémentation

Liste et rôle des fichiers de cette version :

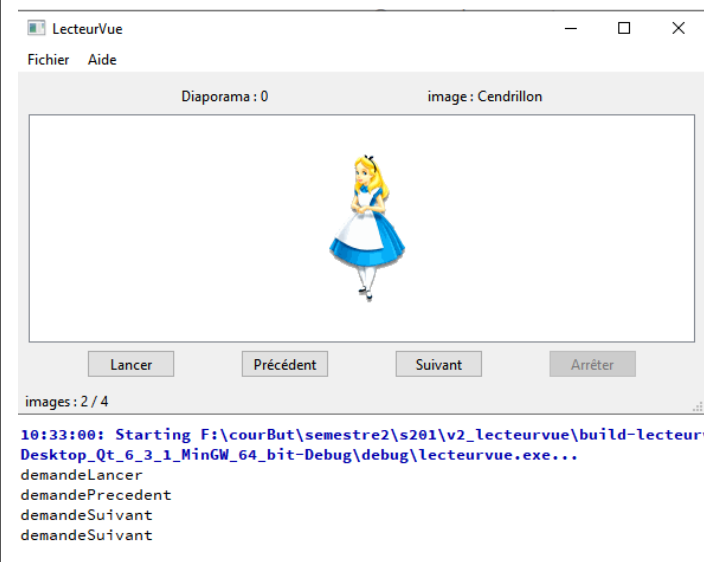
lecteurvue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas Concerne la gestion graphique de l'application.
lecteurvue.cpp	Corps de la classe LecteurVue.
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner
lecteurmodele.h	Spécification de la classe LecteurModele.

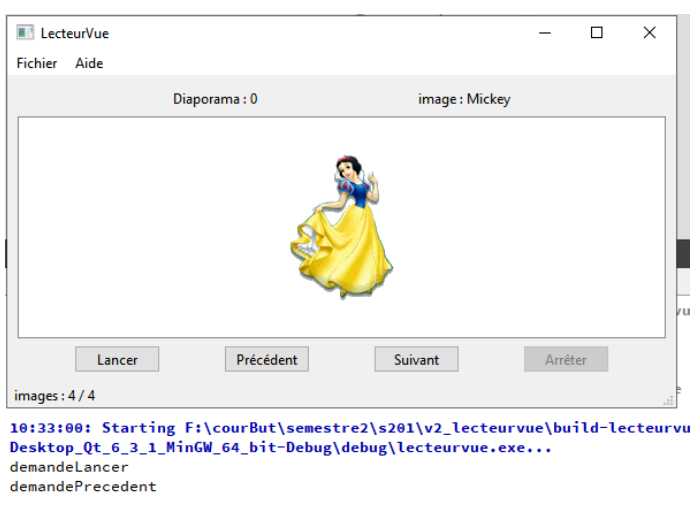
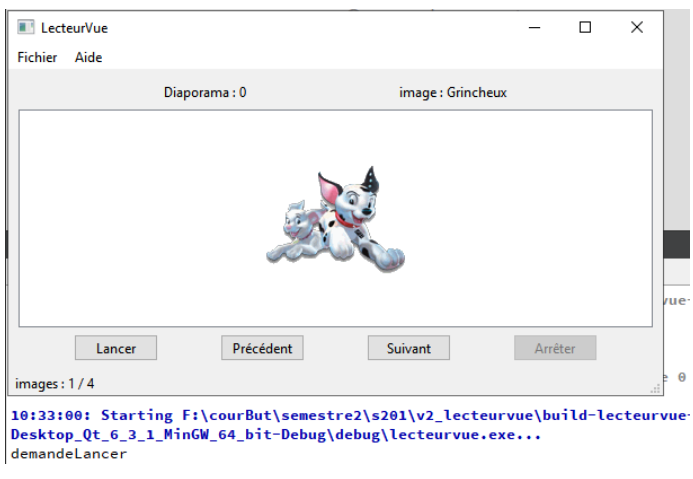
	Est centré sur informations et les actions métier et déconnecté du dialogue utilisateur.
lecteurmodele.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image Permet d'accéder aux diverses informations de l'image, comme le nom, son chemin, sa catégorie, via ses méthodes.
image.cpp	Corps de la classe Image
main.cpp	Permet de créer l'application et la fenêtre principale LecteurVue.

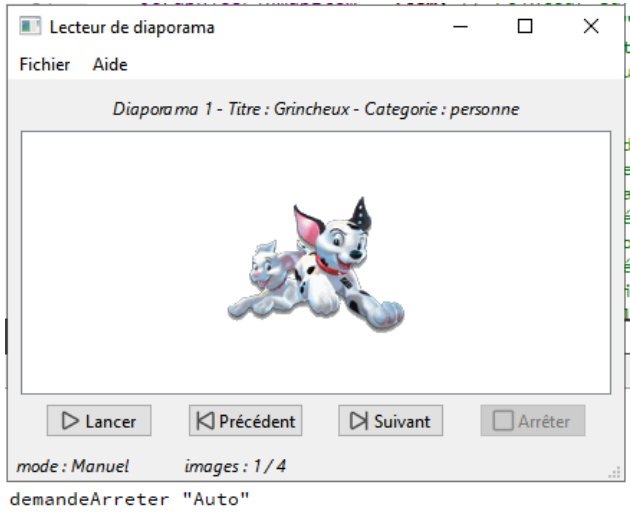
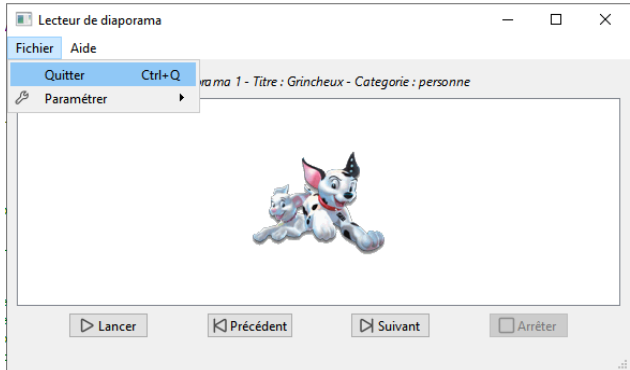
Remarques sur l'implémentation :

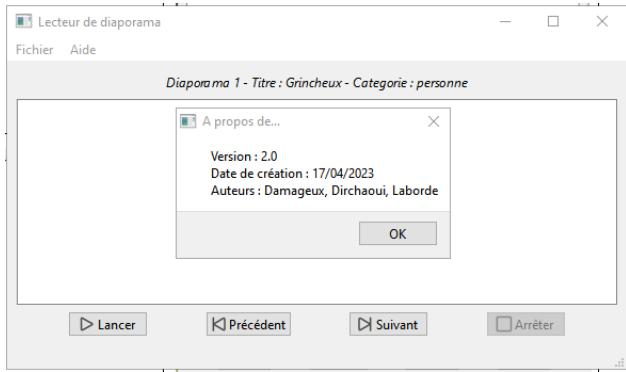
Commenter brièvement les choix importants d'implémentation réalisés, comme par exemple, les signals/slots

5.3.2 Tests (v3)

Opération testé	Description	Comportement attendu	Comportement observé
Passage à l'image suivante	Dans cette version, cette opération change l'image courante affichée et affiche l'image suivante.	Affichage de l'image suivante.	 <pre> 10:33:00: Starting F:\courBut\semestre2\s201\v2_lecteurvue\build-lecteurvue\ Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... demandeLancer demandePrecedent demandeSuivant demandeSuivant </pre>

Passage à l'image précédente	Dans cette version, cette opération change l'image courante affichée et affiche l'image précédente.	Affichage de l'image précédente.	 <pre> 10:33:00: Starting F:\courBut\semestre2\s201\v2_lecteurvue\build-lecteurvue\ Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... demandeLancer demandePrecedent </pre>
Lancement lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action de lancer la lecture du diaporama a été demandée.	Affichage dans la console du message "action lancer le diaporama".	 <pre> 10:33:00: Starting F:\courBut\semestre2\s201\v2_lecteurvue\build-lecteurvue\ Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe... demandeLancer </pre>

Arrêt lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action d'arrêter la lecture du diaporama a été demandée.	Affichage dans la console du message "action arreter le diaporama".	
Quitter l'application	L'appui dans le menu fichier de l'action "Quitter" arrête l'application	Arrêt de l'application, la fenêtre principale se ferme.	

Afficher informations application	L'appui dans le menu aide de l'action "A propos de..." ouvre une boîte de message donnant les informations sur l'application tel que : sa version ; date de création ; les auteurs.	Affichage d'une fenêtre de dialogue contenant les informations sur l'application.	
-----------------------------------	---	---	---

A faire :

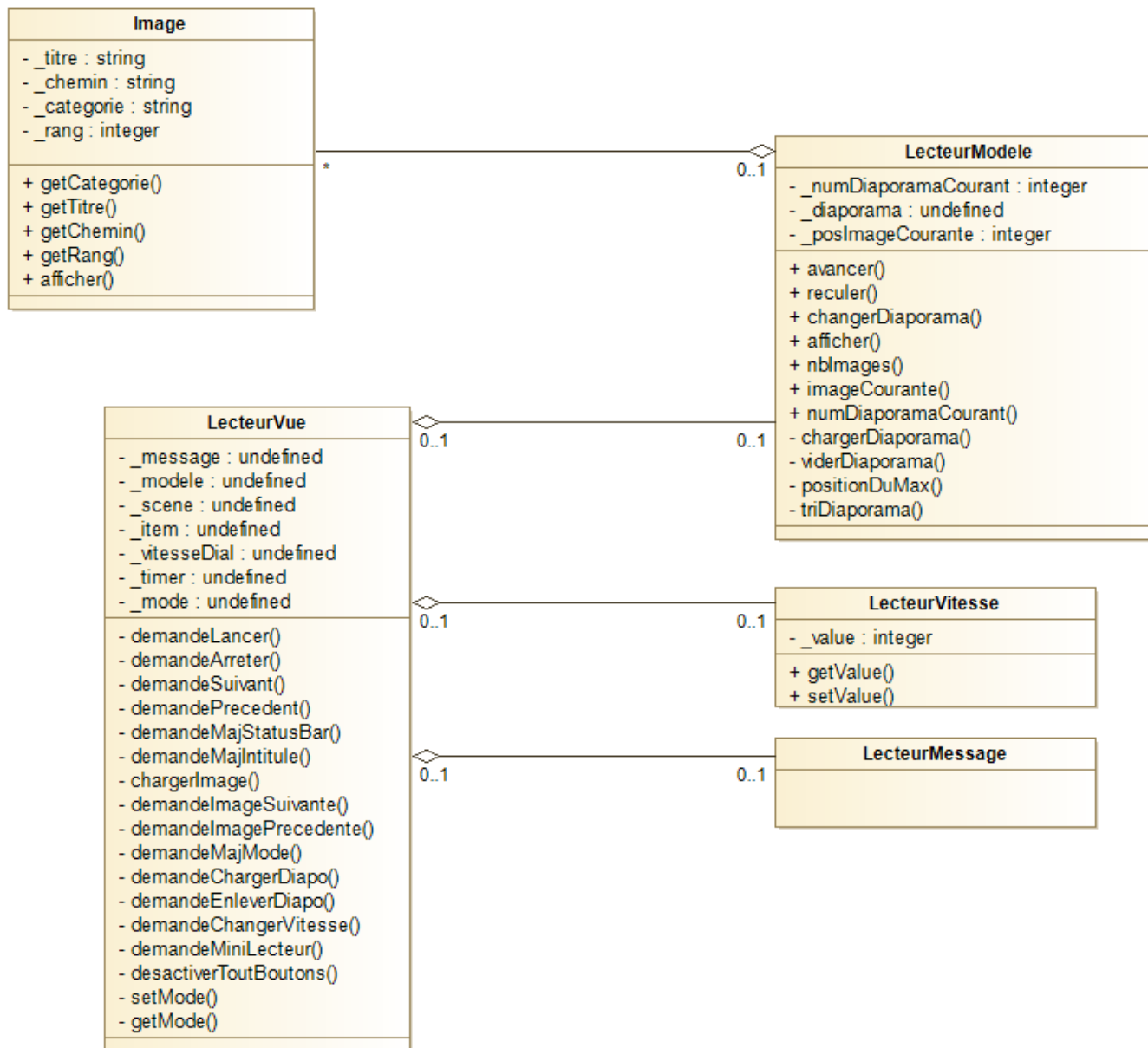
Décrire les tests prévus / réalisés pour montrer :

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

Le comportement fonctionnel de l'application

6. VERSION V4 –

6.1 Diagramme de classes (UML)



`_diaporama` est de type `Diaporama`, nous ne l'avons pas mis sur modelio car le type `Diaporama` n'est pas disponible. `_message` est de type `LecteurMessage`, `_modele` est de type `LecteurModele`, `_scene` est de type `QGraphicsScene` et `_item` est de type `QGraphicsPixmapItem`.
`_timer` est de type `QTimer`. `_mode` est de type `QString`. `_vitesseDial` est de type `LecteurVitesse`.

6.2 Comportement de l'application

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

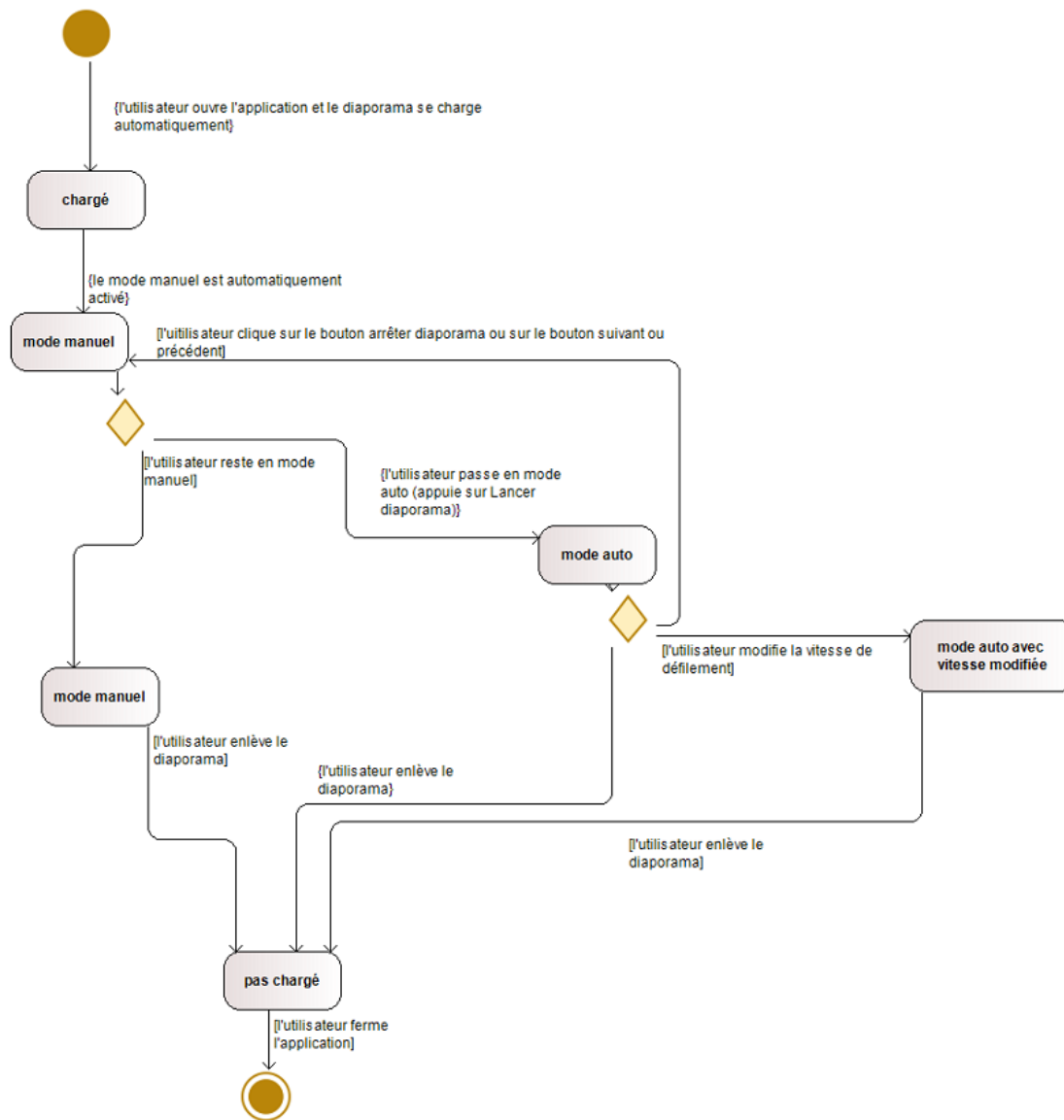


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v4

6.2.2 Dictionnaire des états, événements et Actions (v4)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
Pas chargé	L'application est lancée mais aucun diaporama n'est chargé
chargé	L'utilisateur a chargé un diaporama
Mode manuel	Le diaporama se parcourt grâce aux boutons et c'est l'utilisateur qui doit les activer
Mode auto	Le diaporama se parcourt automatiquement avec la vitesse par défaut
Mode auto avec vitesse choisie	Le diaporama se parcourt automatiquement avec la vitesse choisie par l'utilisateur.
Arrêté	L'utilisateur arrête la lecture du diaporama
enleve	L'utilisateur enlève le diaporama
Fermé	L'utilisateur ferme l'application

Tableau 2 : États du lecteur de diaporamas – v4

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomÉvénement</i>	<i>Signification</i>
L'utilisateur ouvre l'application	L'utilisateur ouvre l'application
L'utilisateur charge son diaporama	L'utilisateur charge un diaporama afin de l'afficher
L'utilisateur ne passe pas en mode auto	L'utilisateur reste en mode manuel pour le parcours du diaporama
L'utilisateur passe en mode auto et ne change pas la vitesse de déroulement	L'utilisateur passe en mode auto avec la vitesse de défilement par défaut
L'utilisateur passe en mode auto et change la vitesse de déroulement	L'utilisateur passe en mode auto avec la vitesse de déroulement choisie
L'utilisateur parcourt le diaporama	L'utilisateur passe les images à l'aide des boutons
Le diaporama se déroule automatiquement	Le diaporama passe les images à la vitesse réglée (par défaut ou par l'utilisateur selon le cas)
L'utilisateur clique sur le bouton arrêter	L'utilisateur repasse en mode manuel
L'utilisateur enlève le diaporama	L'utilisateur enlève le diaporama chargé.

Tableau 3 : Événements faisant changer le diaporama d'état – v4

Description des actions réalisées lors de la traversée des transitions 1er partie

<i>nomAction</i>	<i>Signification</i>
l'application s'ouvre	l'application s'ouvre
le diaporama se charge	le diaporama est chargé dans le lecteur diaporama
l'utilisateur passe en mode manuel	le mode manuel est activé, lecteur attend les interactions de l'utilisateur
l'utilisateur passe au diaporama suivant	le lecteur diaporama passe à l'image suivante
l'utilisateur passe au diaporama précédent	le lecteur diaporama retourne à l'image précédente
l'utilisateur arrête la lecture du diaporama	le lecteur diaporama arrête la lecture du diaporama
l'utilisateur ferme l'application	le lecteur diaporama se ferme

Tableau 4 : Actions à réaliser lors des changements d'état – lecteur de diaporamas v4

Description des actions réalisées lors de la traversée des transitions 2ème partie

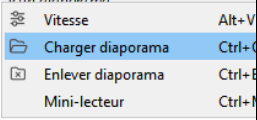
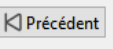
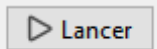
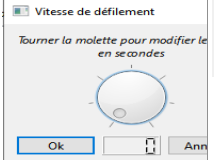
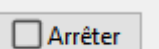
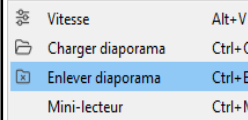
<i>nomAction</i>	<i>Signification</i>
l'application s'ouvre	l'application est ouverte
le diaporama se charge	le lecteur diaporama affiche le diaporama chargé
lecteur passe au mode auto	lecteur diaporama passe en mode automatique
passage lecture auto	une lecture en mode auto du diaporama avec une vitesse normale prédéfinie
passage lecture auto avec le changement de vitesse de déroulement	une lecture en mode auto avec la vitesse de déroulement choisie par l'utilisateur
l'arrêt du diaporama	le lecteur diaporama arrête la lecture du diaporama
diaporama s'enlève	le diaporama est enlevé du lecteur
l'application se ferme	l'application se ferme

6.2.3 Table *T_EtatsEvenementsActions (v4)*

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

2. en ligne : les *états* du lecteur de diaporamas (éventuel état de départ d'une transition)

3. en colonne : les **événements** faisant changer le lecteur d'état (déclencheur d'une transition)
4. 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 <input type="checkbox"/>								
Événement <input type="checkbox"/> nomEtat	L'utilisateur ouvre l'application	l'utilisateur charge son diaporama	l'utilisateur clique sur un des boutons	l'utilisateur passe en mode auto sans changer de vitesse de déroulement	l'utilisateur passe en mode auto et change la vitesse de déroulement	L'utilisateur clique sur le bouton arrêter	L'utilisateur enlève le diaporama	l'utilisateur ferme l'application
<i>pas chargé</i>	l'application s'ouvre							
<i>chargé</i>		le diaporama se charge						
<i>mode manuel</i>			-passage à l'image suivante -passage à l'image précédente	passage des images en mode auto				
<i>mode auto</i>			passage en mode manuel			l'arrêt du diaporama		
<i>mode auto avec vitesse choisie</i>					passage lecture auto avec le changement de vitesse de déroulement	l'arrêt du diaporama		
<i>enlevé</i>							diaporama s'enlève	

Fermé								l'application se ferme
-------	--	--	--	--	--	--	--	------------------------

Tableau 5 : Matrice d'états-transitions du lecteur de diaporamas – v4

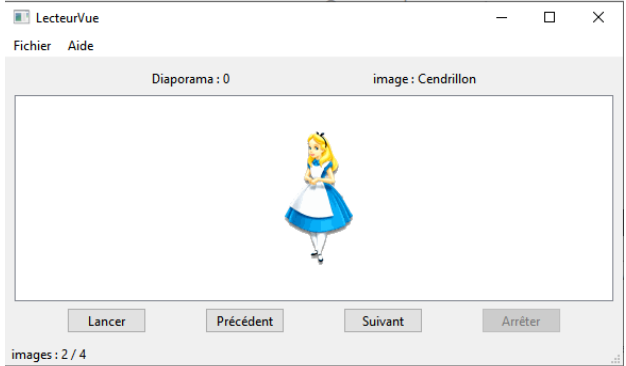
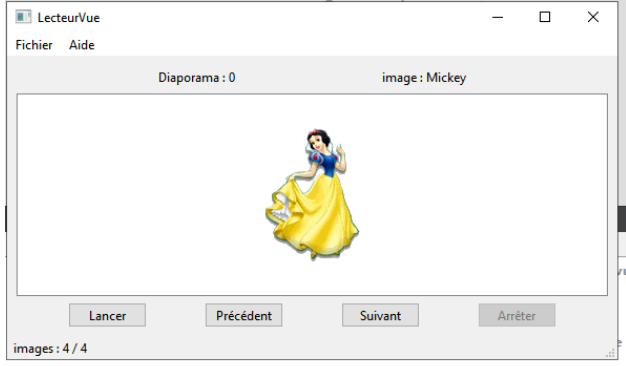
6.3 Implémentation et tests

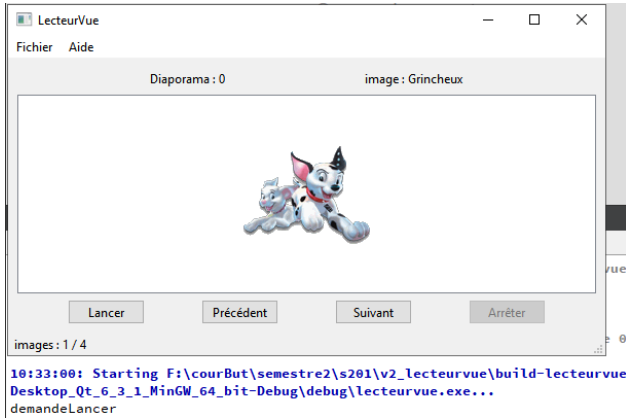
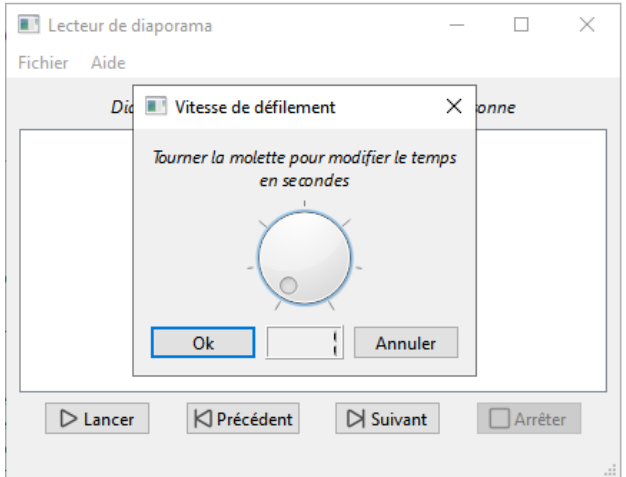
6.3.1 Implémentation

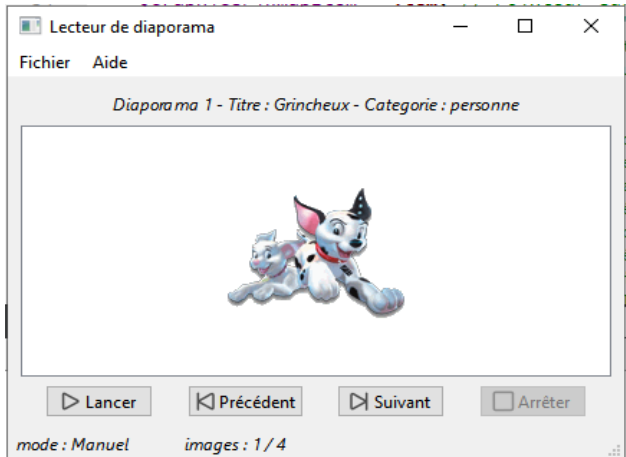
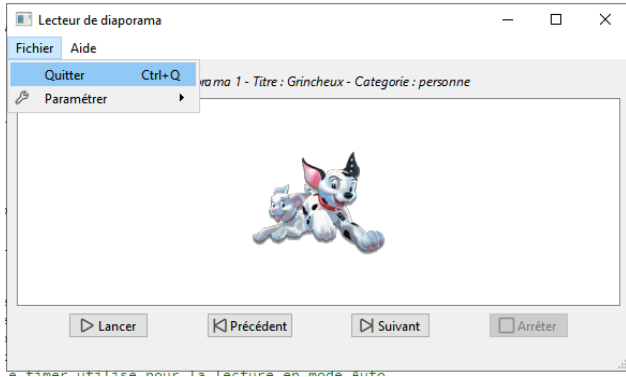
Liste et rôle des fichiers de cette version :

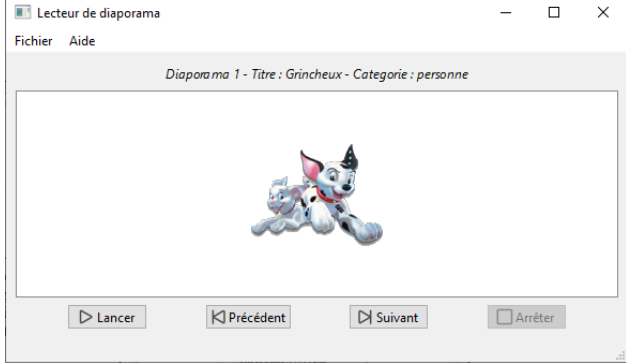
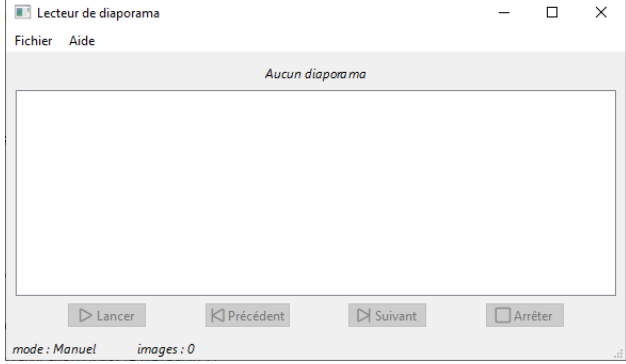
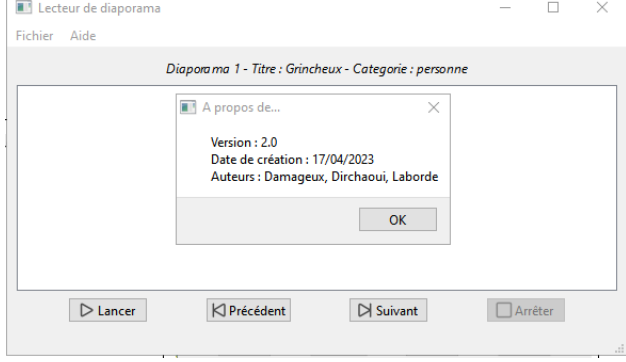
lecteurvue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas Concerne la gestion graphique de l'application.
lecteurvue.cpp	Corps de la classe LecteurVue.
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner
lecteurmodele.h	Spécification de la classe LecteurModele. Est centré sur informations et les actions métier et déconnecté du dialogue utilisateur.
lecteurmodele.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image Permet d'accéder aux diverses informations de l'image, comme le nom, son chemin, sa catégorie, via ses méthodes.
lecteurvitesse.h	Spécification de la classe lecteurvitesse Permet d'accéder aux divers option de la vitesse de défilement de l'image
lecteurvitesse.cpp	Corps de la classe lecteurvitesse
image.cpp	Corps de la classe Image
main.cpp	Permet de créer l'application et la fenêtre principale LecteurVue.
lecteurmessage.h	Spécification de la classe LecteurMessage Permet d'afficher la fenêtre A propos
lecteurmessage.cpp	Corps de la classe LecteurMessage

6.3.2 Tests (v4)

Opération testé	Description	Comportement attendu	Comportement observé
Passage à l'image suivante	Dans cette version, cette opération change l'image courante affichée et affiche l'image suivante.	Affichage de l'image suivante.	 <p>The screenshot shows the LecteurVue application window. The title bar says 'LecteurVue'. Below it are menu items 'Fichier' and 'Aide'. The main area has a status bar at the top showing 'Diaporama : 0' and 'image : Cendrillon'. The central display area shows a cartoon image of Alice. Below the image are four buttons: 'Lancer', 'Précédent', 'Suivant', and 'Arrêter'. At the bottom left, it says 'images : 2 / 4'. Below the window, there is a log of events: '10:33:00: Starting F:\courBut\semestre2\s201\v2_lecteurvue\build-lecteurvue\Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe...', 'demandeLancer', 'demandePrecedent', 'demandeSuivant', and 'demandeSuivant'.</p>
Passage à l'image précédente	Dans cette version, cette opération change l'image courante affichée et affiche l'image précédente.	Affichage de l'image précédente.	 <p>The screenshot shows the LecteurVue application window. The title bar says 'LecteurVue'. Below it are menu items 'Fichier' and 'Aide'. The main area has a status bar at the top showing 'Diaporama : 0' and 'image : Mickey'. The central display area shows a cartoon image of Snow White. Below the image are four buttons: 'Lancer', 'Précédent', 'Suivant', and 'Arrêter'. At the bottom left, it says 'images : 4 / 4'. Below the window, there is a log of events: '10:33:00: Starting F:\courBut\semestre2\s201\v2_lecteurvue\build-lecteurvue\Desktop_Qt_6_3_1_MinGW_64_bit-Debug\debug\lecteurvue.exe...', 'demandeLancer', and 'demandePrecedent'.</p>

Lancement lecture du diaporama en mode “Auto”	Dans cette version,cette opération permet le passage des image en mode automatique	Affichage dans la console du message “action lancer le diaporama”.	
changement de vitesse du diaporama en mode “Auto”	Dans cette version,cette opération permet de changer la vitesse de défilement des image en mode automatique	Affichage dans la console du message “Vitesse”.	

Arrêt lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche une message de debug informant que l'action d'arrêter la lecture du diaporama a été demandée.	Affichage dans la console du message "action arrêter le diaporama", L'image courante reste affichée, la lecture du diaporama pense en mode "Manuel"	
Quitter l'application	L'appui dans le menu fichier de l'action "Quitter" arrête l'application	Arrêt de l'application, la fenêtre principale se ferme.	

Charger diaporama	L'appui dans le menu fichier, puis paramétrer puis charger diaporama, charge le diaporama et affiche l'image courante	Le diaporama se charge, l'image courante s'affiche.	
Enlever diaporama	L'appui dans le menu fichier, puis paramétrer puis enlever diaporama, enlève le diaporama et l'image courante.	Le diaporama s'enlève et aucune image n'est affichée. Le lecteur est vide à nouveau.	
Afficher informations application	L'appui dans le menu aide de l'action "A propos de..." ouvre une boîte de message donnant les informations sur l'application tel que : sa version ; date de création ; les auteurs.	Affichage d'une fenêtre de dialogue contenant les informations sur l'application.	

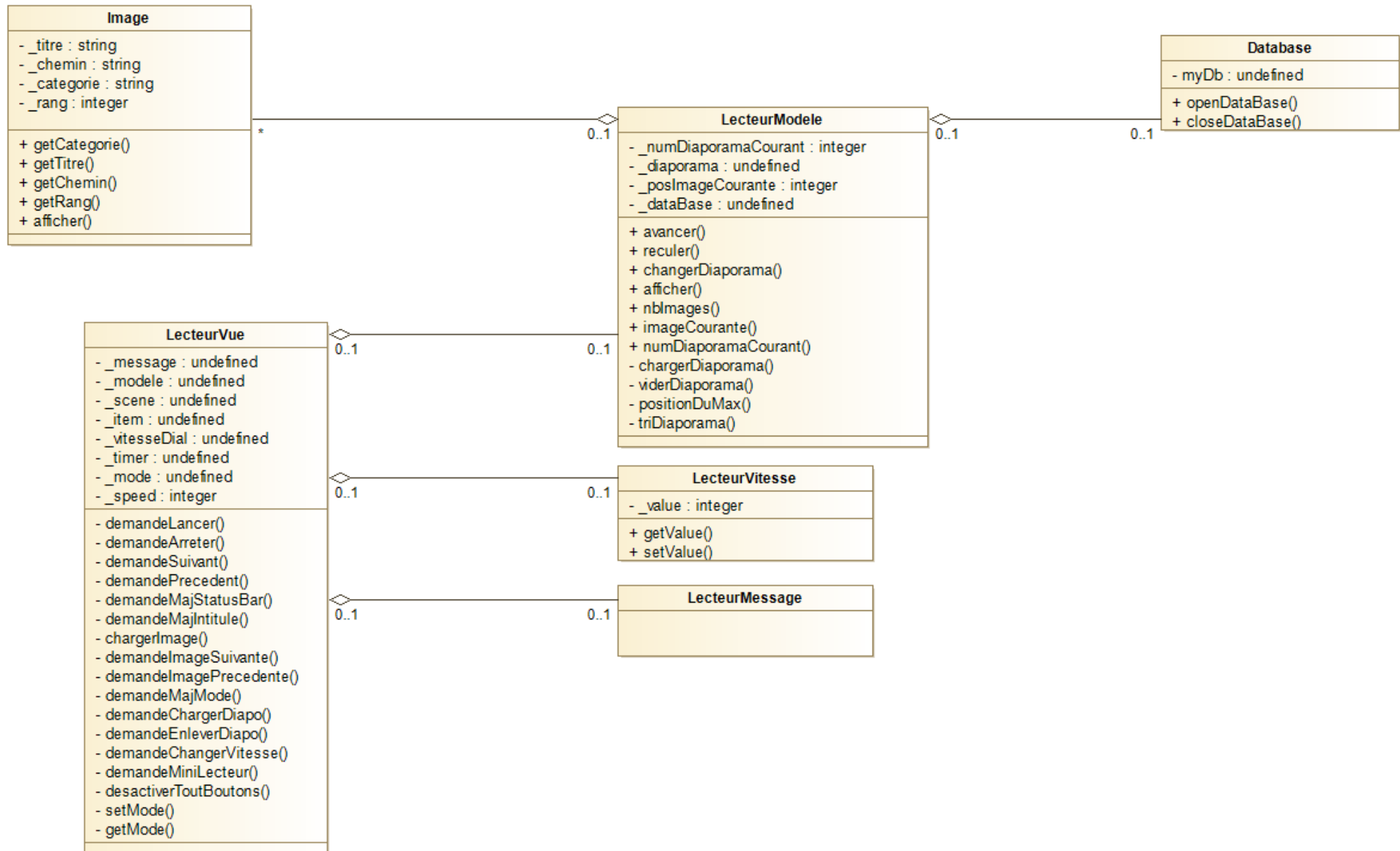
A faire :

Décrire les tests prévus / réalisés pour montrer :

5. Le comportement de l'interface non lié aux aspects fonctionnels du programme
6. Le comportement de l'interface liée aux aspects fonctionnels du programme
7. Le comportement fonctionnel de l'application

7. VERSION V5 –

7.1 Diagramme de classes (UML)



`_diaporama` est de type `Diaporama`, nous ne l'avons pas mis sur modelio car le type `Diaporama` n'est pas disponible. `_message` est de type `LecteurMessage`, `_modele` est de type `LecteurModele`, `_scene` est de type `QGraphicsScene` et `_item` est de type `QGraphicsPixmapItem`.
`_timer` est de type `QTimer`. `_mode` est de type `QString`. `_vitesseDial` est de type `LecteurVitesse`. `myDb` est un `QSqlDatabase`. `_dataBase` est de type `Database`
`Database`

7.2 Comportement de l'application

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

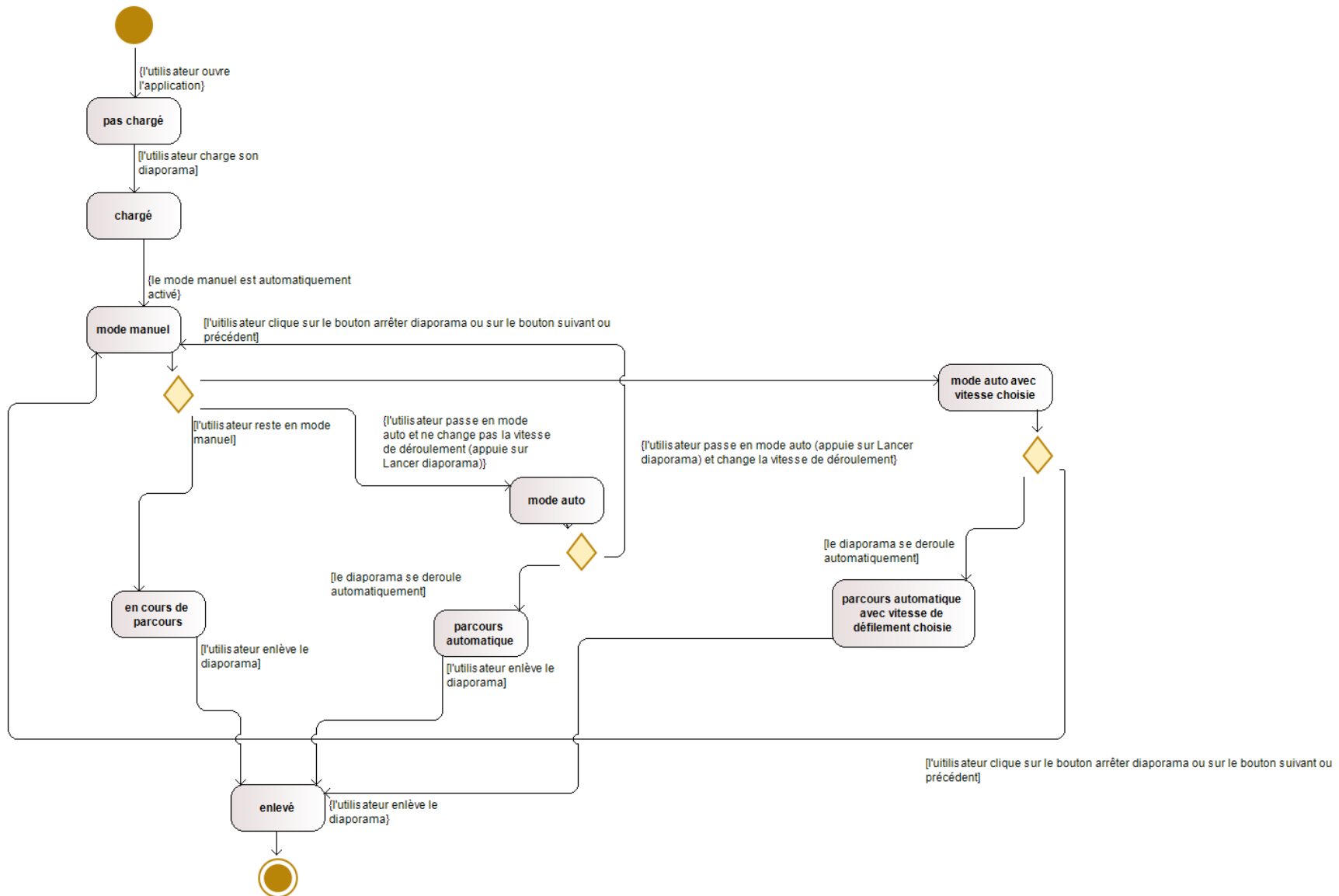


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v5

7.2.2 Dictionnaire des états, événements et Actions (v5)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
Pas chargé	L'application est lancée mais aucun diaporama n'est chargé
chargé	L'utilisateur a chargé un diaporama
Mode manuel	Le diaporama se parcourt grâce aux boutons et c'est l'utilisateur qui doit les activer
Mode auto	Le diaporama se parcourt automatiquement avec la vitesse par défaut
Mode auto avec vitesse choisie	Le diaporama se parcourt automatiquement avec la vitesse choisie par l'utilisateur.
en cours de parcours	L'utilisateur est en train de parcourir le diaporama en mode manuel
parcours automatique	Le diaporama est en cours de parcours automatique et passe les images à la vitesse par défaut.
parcours automatique	Le diaporama est en cours de parcours automatique et passe les images à la vitesse choisie par l'utilisateur.
enlevé	L'utilisateur a enlevé son diaporama

Tableau 2 : États du lecteur de diaporamas – v5

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomEvénement</i>	<i>Signification</i>
L'utilisateur ouvre l'application	L'utilisateur ouvre l'application
L'utilisateur charge son diaporama	L'utilisateur charge un diaporama afin de l'afficher
L'utilisateur ne passe pas en mode auto	L'utilisateur reste en mode manuel pour le parcours du diaporama
L'utilisateur passe en mode auto et ne change pas la vitesse de déroulement	L'utilisateur passe en mode auto avec la vitesse de défilement par défaut
L'utilisateur passe en mode auto et change la vitesse de déroulement	L'utilisateur passe en mode auto avec la vitesse de déroulement choisie
L'utilisateur parcourt le diaporama	L'utilisateur passe les images à l'aide des boutons
Le diaporama se déroule automatiquement	Le diaporama passe les images à la vitesse réglée (par défaut ou par l'utilisateur selon le cas)
L'utilisateur clique sur le bouton arrêter diaporama ou sur le bouton suivant ou précédent	L'utilisateur passe en mode manuel
L'utilisateur enlève le diaporama	L'utilisateur enlève le diaporama chargé.

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

Description des actions réalisées lors de la traversée des transitions 1er partie

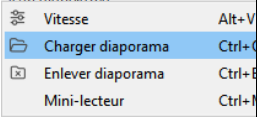
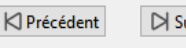
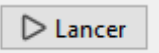
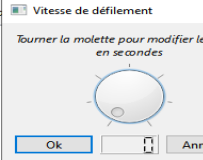
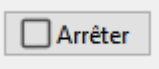
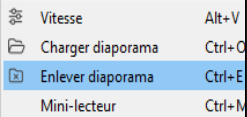
<i>nomAction</i>	<i>Signification</i>
l'application s'ouvre	l'application s'ouvre
le diaporama se charge	le diaporama est chargé dans le lecteur diaporama
l'utilisateur passe en mode manuel	le mode manuel est activé, lecteur attends les interaction de l'utilisateur
l'utilisateur passe au diaporama suivant	le lecteur diaporama passe à l'image suivante
l'utilisateur passe au diaporama précédent	le lecteur diaporama retourne à l'image précédente
l'utilisateur arrête la lecture du diaporama	le lecteur diaporama arrête la lecture du diaporama

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

Description des actions réalisées lors de la traversée des transitions 2ème partie

<i>nomAction</i>	<i>Signification</i>
L'utilisateur ouvre l'application	l'application s'ouvre
L'utilisateur charge son diaporama	le lecteur diaporama affiche le diaporama chargé
l'utilisateur passe en mode auto	lecteur diaporama passe en mode automatique
L'utilisateur ne change pas la vitesse de déroulement	lecteur diaporama
L'utilisateur passe en mode auto et change la vitesse de déroulement	L'utilisateur passe en mode auto avec la vitesse de déroulement choisie
l'utilisateur arrête la lecture du diaporama	l'utilisateur arrête la lecture auto du diaporama en appuyant sur le bouton arrêter

7.2.3 Table *T_EtatsEvenementsActions (v5)*

<i>Élément graphique pregnant en charge cet événement</i> <input type="checkbox"/>								
<i>Événement</i> <input type="checkbox"/> <i>nomEtat</i>	L'utilisateur ouvre l'application	l'utilisateur charge son diaporama	l'utilisateur clique sur un des boutons	l'utilisateur passe en mode auto sans changer de vitesse de déroulement	l'utilisateur passe en mode auto et change la vitesse de déroulement	L'utilisateur clique sur le bouton arrêter	L'utilisateur enlève le diaporama	l'utilisateur ferme l'application
<i>pas chargé</i>	l'application s'ouvre							
<i>chargé</i>		le diaporama se charge						
<i>mode manuel</i>			-passage à l'image suivante -passage à l'image précédente	passage des images en mode auto				
<i>mode auto</i>			passage en mode manuel			l'arrêt du diaporama		
<i>mode auto avec vitesse choisie</i>					passage lecture auto avec le changement de vitesse de déroulement	l'arrêt du diaporama		
<i>enlevé</i>							diaporama s'enlève	
<i>Fermé</i>								l'application se

								ferme
--	--	--	--	--	--	--	--	-------

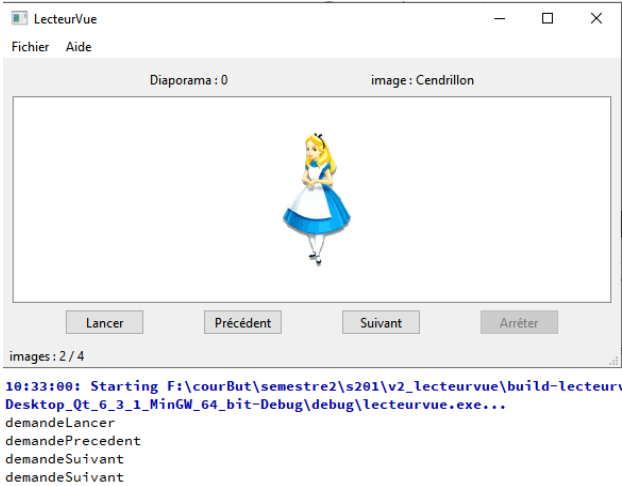
7.3 Implémentation et tests

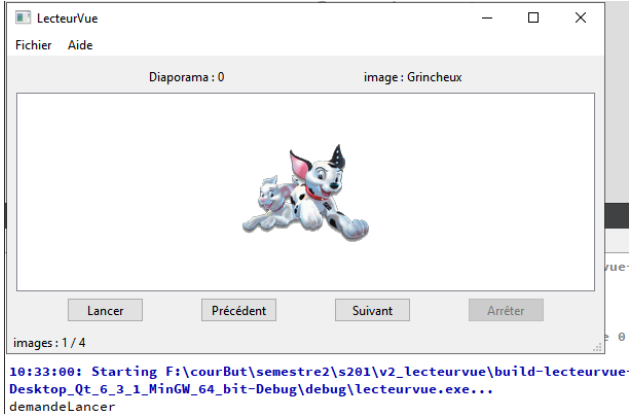
7.3.1 Implémentation

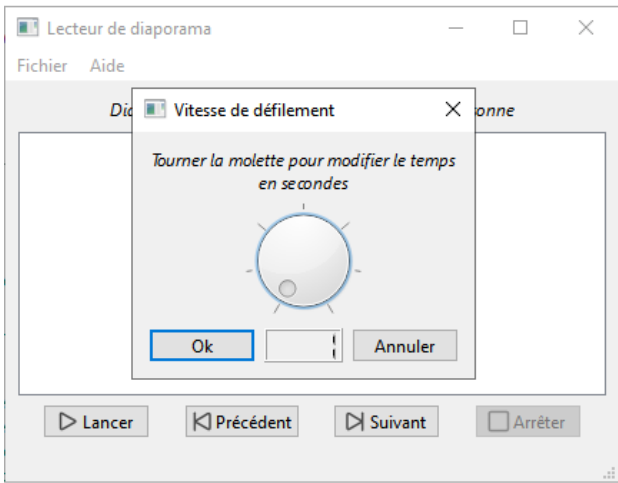
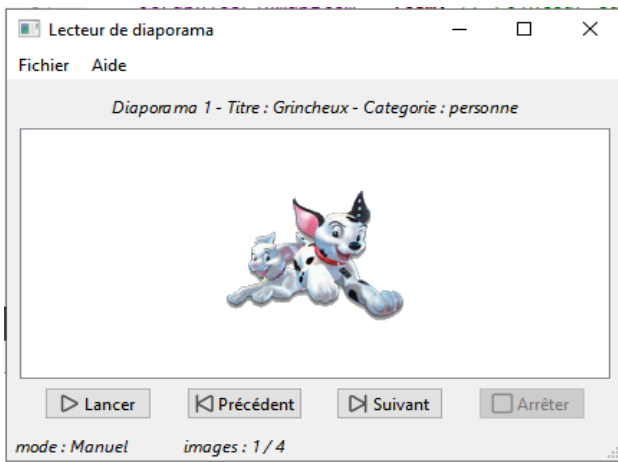
Liste et rôle des fichiers de cette version :

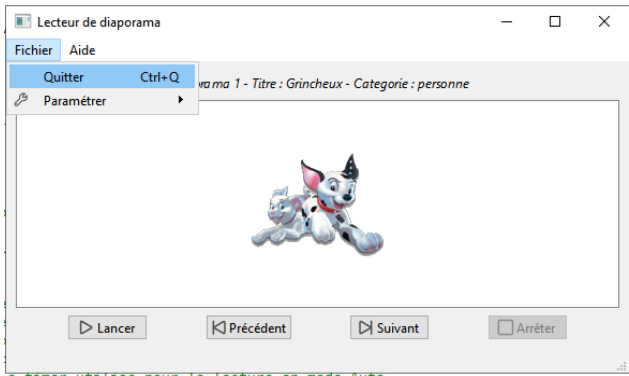
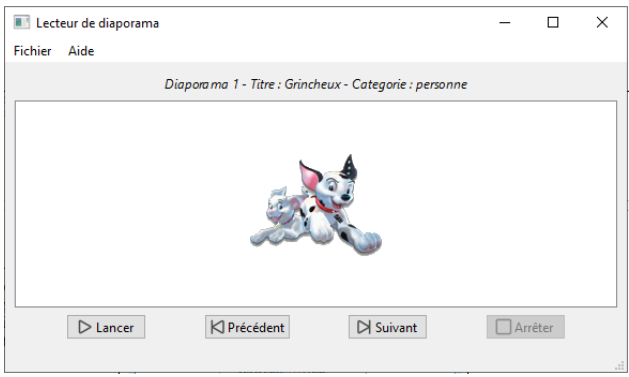
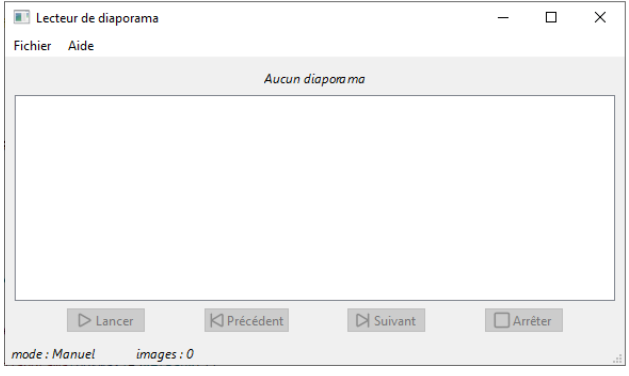
lecteurvue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas Concerne la gestion graphique de l'application.
lecteurvue.cpp	Corps de la classe LecteurVue.
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner
lecteurmodele.h	Spécification de la classe LecteurModele. Est centré sur informations et les actions métier et déconnecté du dialogue utilisateur.
lecteurmodele.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image Permet d'accéder aux diverses informations de l'image, comme le nom, son chemin, sa catégorie, via ses méthodes.
lecteurvitesse.h	Spécification de la classe lecteurvitesse Permet d'accéder aux divers option de la vitesse de défilement de l'image
lecteurvitesse.cpp	Corps de la classe lecteurvitesse
image.cpp	Corps de la classe Image
main.cpp	Permet de créer l'application et la fenêtre principale LecteurVue.
lecteurmessage.h	Spécifications de la classe LecteurMessage Permet d'afficher la fenêtre A propos
lecteurmessage.cpp	Corps de la classe LecteurMessage
database.h	Spécifications de la classe Database (créer un connexion vers la base de donnée)
database.cpp	Corps de la classe Database

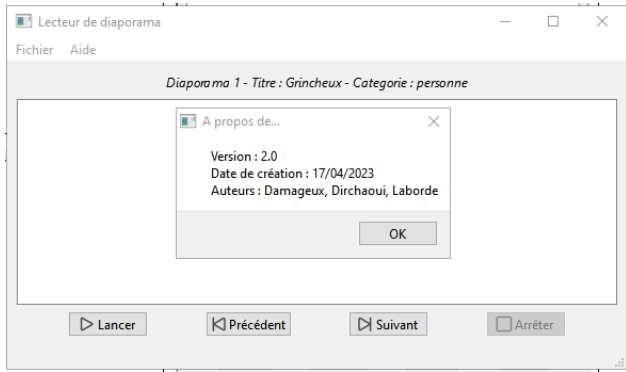
7.3.2 Tests (v5)

Opération testé	Description	Comportement attendu	Comportement observé
Passage à l'image suivante	Dans cette version, cette opération change l'image courante affichée et affiche l'image suivante.	Affichage de l'image suivante.	

Passage à l'image précédente	Dans cette version, cette opération change l'image courante affichée et affiche l'image précédente.	Affichage de l'image précédente.	
Lancement lecture du diaporama en mode "Auto"	Dans cette version, cette opération permet le passage des image en mode automatique	Affichage dans la console du message "action lancer le diaporama".	

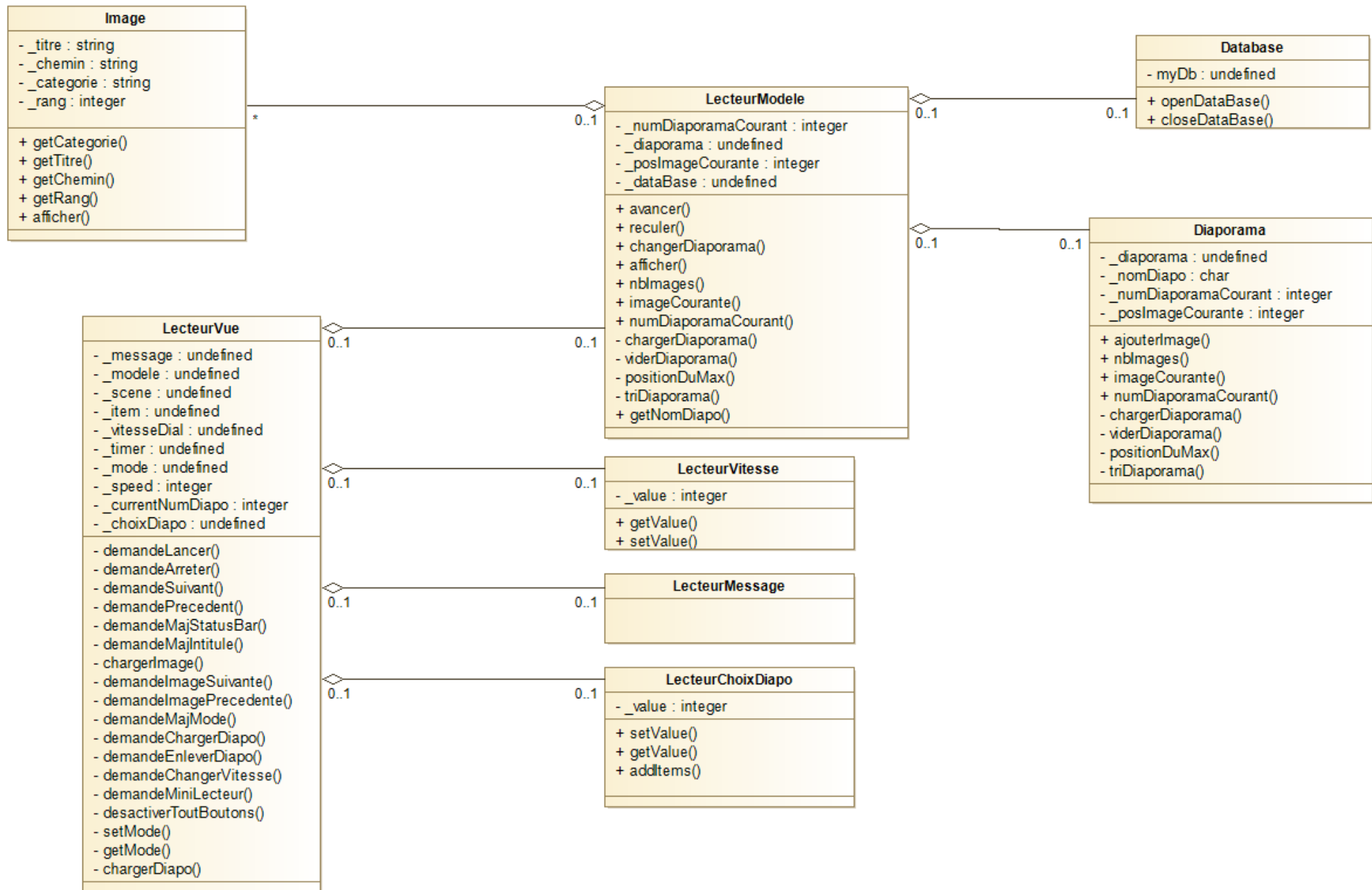
<p>changement de vitesse du diaporama en mode "Auto"</p>	<p>Dans cette version, cette opération permet de changer la vitesse de défilement des images en mode automatique</p>	<p>Affichage dans la console du message "Vitesse".</p>	
<p>Arrêt lecture du diaporama en mode "Auto"</p>	<p>Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche un message de debug informant que l'action d'arrêter la lecture du diaporama a été demandée.</p>	<p>Affichage dans la console du message "action arrêter le diaporama", L'image courante reste affichée, la lecture du diaporama passe en mode "Manuel"</p>	

Quitter l'application	L'appui dans le menu fichier de l'action "Quitter" arrête l'application	Arrêt de l'application, la fenêtre principale se ferme.	
Charger diaporama	L'appui dans le menu fichier, puis paramétrer puis charger diaporama, charge le diaporama et affiche l'image courante	Le diaporama se charge, l'image courante s'affiche.	
Enlever diaporama	L'appui dans le menu fichier, puis paramétrer puis enlever diaporama, enlève le diaporama et l'image courante.	Le diaporama s'enlève et aucune image n'est affichée. Le lecteur est vide à nouveau.	

Afficher informations application	L'appui dans le menu aide de l'action "A propos de..." ouvre une boîte de message donnant les informations sur l'application tel que : sa version ; date de création ; les auteurs.	Affichage d'une fenêtre de dialogue contenant les informations sur l'application.	
-----------------------------------	---	---	---

8. VERSION V6 –

8.1 Diagramme de classes (UML)



`_diaporama` est de type `Diaporama`, nous ne l'avons pas mis sur modelio car le type `Diaporama` n'est pas disponible. `_message` est de type `LecteurMessage`, `_modele` est de type `LecteurModele`, `_scene` est de type `QGraphicsScene` et `_item` est de type `QGraphicsPixmapItem`.
`_timer` est de type `QTimer`. `_mode` est de type `QString`. `_vitesseDial` est de type `LecteurVitesse`. `myDb` est un `QSqlDatabase`. `_dataBase` est de type `Database`.
`_diaporama` est de type `TableauImages`. `_choixDiapo` est de type `LecteurChoixDiapo`.

8.2 Comportement de l'application

8.2.1 Diagramme états-transitions-actions du lecteur de diaporamas (v6)

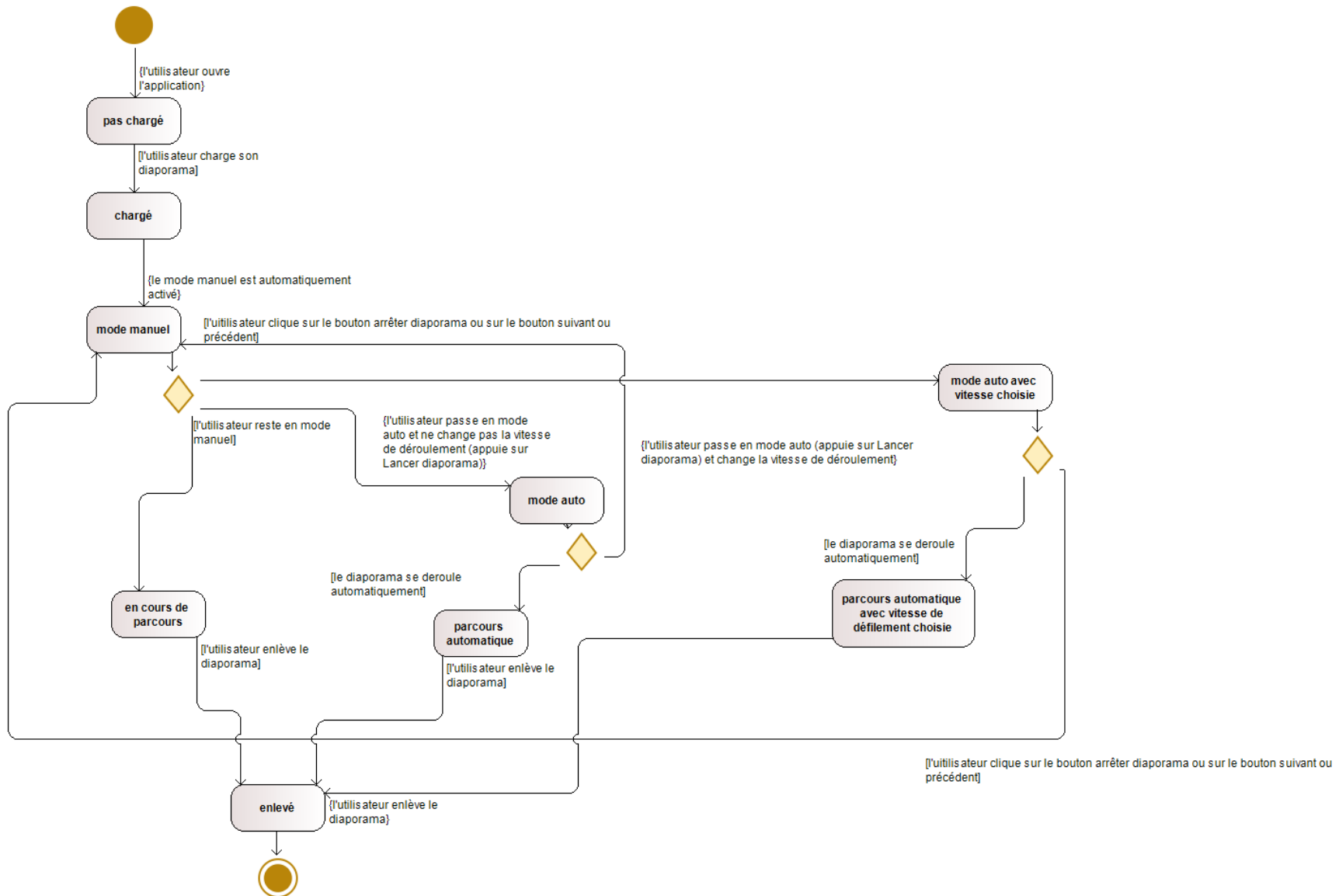


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v6

8.2.2 Dictionnaire des états, événements et Actions (v5)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
Pas chargé	L'application est lancée mais aucun diaporama n'est chargé
chargé	L'utilisateur a chargé un diaporama
Mode manuel	Le diaporama se parcourt grâce aux boutons et c'est l'utilisateur qui doit les activer
Mode auto	Le diaporama se parcourt automatiquement avec la vitesse par défaut
Mode auto avec vitesse choisie	Le diaporama se parcourt automatiquement avec la vitesse choisie par l'utilisateur.
en cours de parcours	L'utilisateur est en train de parcourir le diaporama en mode manuel
parcours automatique	Le diaporama est en cours de parcours automatique et passe les images à la vitesse par défaut.
parcours automatique	Le diaporama est en cours de parcours automatique et passe les images à la vitesse choisie par l'utilisateur.
enlevé	L'utilisateur a enlevé son diaporama

Tableau 2 : États du lecteur de diaporamas – v6

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomEvénement</i>	<i>Signification</i>
L'utilisateur ouvre l'application	L'utilisateur ouvre l'application
L'utilisateur charge son diaporama	L'utilisateur charge un diaporama afin de l'afficher
L'utilisateur ne passe pas en mode auto	L'utilisateur reste en mode manuel pour le parcours du diaporama
L'utilisateur passe en mode auto et ne change pas la vitesse de déroulement	L'utilisateur passe en mode auto avec la vitesse de défilement par défaut
L'utilisateur passe en mode auto et change la vitesse de déroulement	L'utilisateur passe en mode auto avec la vitesse de déroulement choisie
L'utilisateur parcourt le diaporama	L'utilisateur passe les images à l'aide des boutons
Le diaporama se déroule automatiquement	Le diaporama passe les images à la vitesse réglée (par défaut ou par l'utilisateur selon le cas)
L'utilisateur clique sur le bouton arrêter diaporama ou sur le bouton suivant ou précédent	L'utilisateur passe en mode manuel
L'utilisateur enlève le diaporama	L'utilisateur enlève le diaporama chargé.

Tableau 3 : Événements faisant changer le diaporama d'état – v6

Description des actions réalisées lors de la traversée des transitions 1er partie

<i>nomAction</i>	<i>Signification</i>
l'application s'ouvre	l'application s'ouvre
le diaporama se charge	le diaporama est chargé dans le lecteur diaporama
l'utilisateur passe en mode manuel	le mode manuel est activé, lecteur attend les interactions de l'utilisateur
l'utilisateur passe au diaporama suivant	le lecteur diaporama passe à l'image suivante
l'utilisateur passe au diaporama précédent	le lecteur diaporama retourne à l'image précédente
l'utilisateur arrête la lecture du diaporama	le lecteur diaporama arrête la lecture du diaporama

Tableau 4 : Actions à réaliser lors des changements d'état – lecteur de diaporamas v6

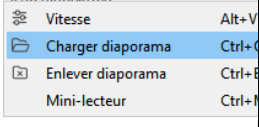
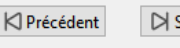
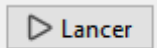
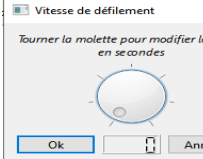
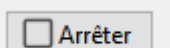
Description des actions réalisées lors de la traversée des transitions 2ème partie

<i>nomAction</i>	<i>Signification</i>
L'utilisateur ouvre l'application	l'application s'ouvre
L'utilisateur charge son diaporama	le lecteur diaporama affiche le diaporama chargé
l'utilisateur passe en mode auto	lecteur diaporama passe en mode automatique
L'utilisateur ne change pas la vitesse de déroulement	lecteur diaporama
L'utilisateur passe en mode auto et change la vitesse de déroulement	L'utilisateur passe en mode auto avec la vitesse de déroulement choisie
l'utilisateur arrête la lecture du diaporama	l'utilisateur arrête la lecture auto du diaporama en appuyant sur le bouton arrêter

8.2.3 Table *T_EtatsEvenementsActions* (v6)

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 <input type="checkbox"/>								
Événement <input type="checkbox"/> nomEtat	L'utilisateur ouvre l'application	l'utilisateur charge son diaporama	l'utilisateur clique sur un des boutons	l'utilisateur passe en mode auto sans changer de vitesse de déroulement	l'utilisateur passe en mode auto et change la vitesse de déroulement	L'utilisateur clique sur le bouton arrêter	L'utilisateur enlève le diaporama	l'utilisateur ferme l'application
pas chargé	l'application s'ouvre							
chargé		le diaporama se charge						
mode manuel			-passage à l'image suivante -passage à l'image précédente	passage des images en mode auto				
mode auto			passage en mode manuel			l'arrêt du diaporama		
mode auto avec vitesse choisie					passage lecture auto avec le changement de vitesse de déroulement	l'arrêt du diaporama		

enlevé							diaporama s'enlève	
Fermé								l'application se ferme

Tableau 5 : Matrice d'états-transitions du lecteur de diaporamas – v6

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

8.3 **Implémentation et tests**

8.3.1 **Implémentation**

Liste et rôle des fichiers de cette version :

lecteurvue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas Concerne la gestion graphique de l'application.
lecteurvue.cpp	Corps de la classe LecteurVue.
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner
lecteurmodele.h	Spécification de la classe LecteurModele. Est centré sur informations et les actions métier et déconnecté du dialogue utilisateur.
lecteurmodele.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image Permet d'accéder aux diverses informations de l'image, comme le nom, son chemin, sa catégorie, via ses méthodes.
lecteurvitesse.h	Spécification de la classe lecteurvitesse Permet d'accéder aux divers option de la vitesse de défilement de l'image
lecteurvitesse.cpp	Corps de la classe lecteurvitesse
image.cpp	Corps de la classe Image
main.cpp	Permet de créer l'application et la fenêtre principale LecteurVue.

lecteurmessage.h	Spécifications de la classe LecteurMessage Permet d'afficher la fenêtre A propos
lecteurmessage.cpp	Corps de la classe LecteurMessage
database.h	Spécifications de la classe Database (créer une connexion vers la base de donnée)
database.cpp	Corps de la classe Database

8.3.2 Tests (v6)

Opération testé	Description	Comportement attendu	Comportement observé
Passage à l'image suivante	Dans cette version, cette opération change l'image courante affichée et affiche l'image suivante.	Affichage de l'image suivante.	
Passage à l'image précédente	Dans cette version, cette opération change l'image courante affichée et affiche l'image précédente.	Affichage de l'image précédente.	
Lancement lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche un message de debug informant que l'action de lancer la lecture du diaporama a été demandée.	Affichage dans la console du message "action lancer le diaporama".	
Arrêt lecture du diaporama en mode "Auto"	Dans cette version, aucune opération métier ne sera déclenchée, cette opération déclenche un message de debug informant que l'action d'arrêter la lecture du diaporama a été demandée.	Affichage dans la console du message "action arreter le diaporama".	

Quitter l'application	L'appui dans le menu fichier de l'action "Quitter" arrête l'application	Arrêt de l'application, la fenêtre principale se ferme.	
Afficher informations application	L'appui dans le menu aide de l'action "A propos de..." ouvre une boîte de message donnant les informations sur l'application tel que : sa version ; date de création ; les auteurs.	Affichage d'une fenêtre de dialogue contenant les informations sur l'application.	

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 programme
- Le comportement fonctionnel de l'application

9. BILAN

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

Temps global de travail (pour le groupe)

Apprentissages majeurs

Difficultés majeures

Points positifs / négatifs de l'activité