

Saé 2.01 – Développement d'une application

Lecteur de diaporamas – Dossier d'Analyse et conception

URL VERS GITHUB : <https://github.com/Lucas-kiss/S2.01>

Dubos Lucie TD 2 Tp3
KISS Lucas TD 2 Tp3

1. Compléments de spécifications externes.

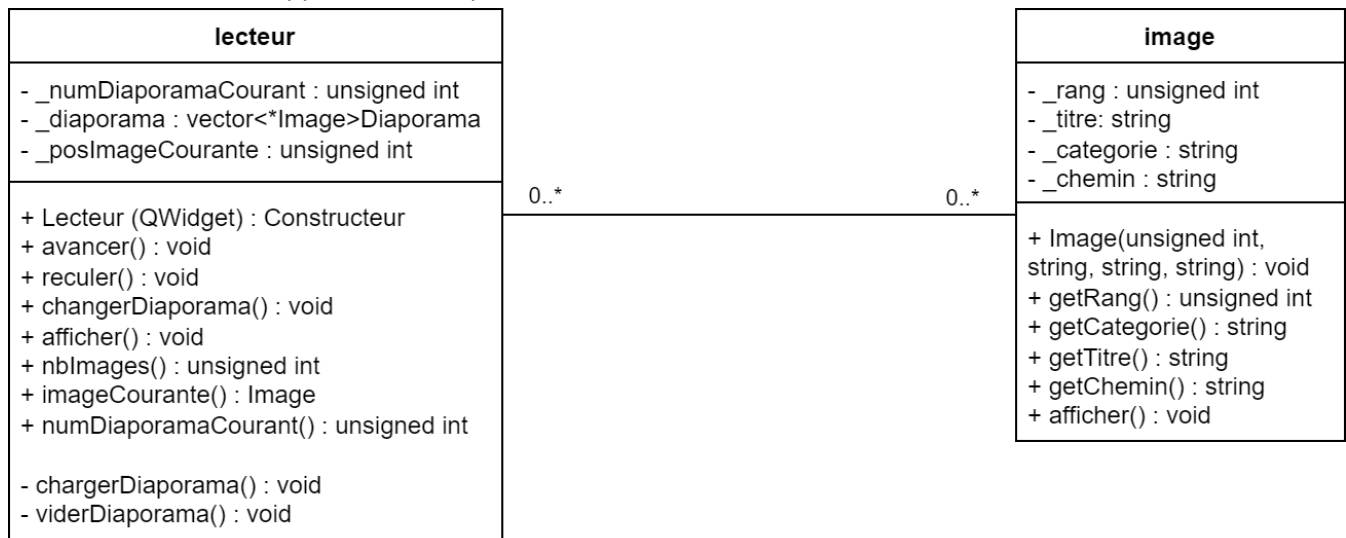
Aucune

2. Scénarios

Cas d'utilisation	Lire un diaporama	
Acteur primaire	Utilisateur	
Système	Lecteur de diaporamas	
Acteur secondaire		
Niveau	Objectif utilisateur	
Préconditions	L'application est lancée	
Opérations	Utilisateur	Système
1	L'utilisateur clique sur Paramètres	
2	L'utilisateur clique sur Charger diaporama	
3		Le système demande de choisir un diaporama
4	L'utilisateur choisi un diaporama	
5		Le système affiche l'image au rang 1 du diaporama
6	L'utilisateur clique sur suivant	
7		Le système affiche l'image suivante
8	L'utilisateur clique sur précédent	
9		Le système affiche l'image précédente
10	L'utilisateur clique sur quitter dans les items de menu quitter	
11		Le système ferme l'application
Extension		
6.A	L'utilisateur clique sur suivant	
6.A.1	L'utilisateur clique sur lancer diaporama ce qui active le mode auto	
6.A.2		Le système affiche les images les unes après les autres à une vitesse prédéfinie
6.A.3	L'utilisateur clique sur l'item de menu changer de vitesse	
6.A.4		Le système affiche une fenêtre de dialogue avec une saisie pour entrer la vitesse choisie
6.A.5	L'utilisateur saisie la vitesse souhaitée et valide	
6.A.6		Le système affiche les images les unes après les autres à la vitesse saisie par l'utilisateur
6.A.7	L'utilisateur clique sur Arrêter diaporama	
6.A.8		Le système arrête le mode automatique et repasse en mode manuel
6.A.9	Retour à l'étape 6	

3. Diagramme de classe (UML)

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



(b) Dictionnaire des éléments pour chaque classe

Classe lecteur			
Nom attribut	Signification	Type	Exemple
_numDiaporamaCourant	Le numéro courant du diaporama manipuler	unsigned int	1
_diaporama	Quelle diaporama est en cours de manipulation	vector<*Image>Diaporama	_diaporama
_posImageCourante	La position de l'image courante	unsigned int	3

Tableau 2 : Dictionnaire des éléments - Classe lecteur

Classe Image			
Nom attribut	Signification	Type	Exemple
_rang	Position de l'image dans le diaporama	unsigned int	1
_titre	Nom de l'image	chaîne de caractere	"Mickey mouse"
_categorie	Catégorie de l'image	chaîne de caractere	"Personne"
_chemin	Chemin vers où se trouve l'image	chaîne de caractere	"F:/Documents/IUT/disney_19.gif"

Tableau 3 : Dictionnaire des éléments - Classe image

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

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 nblImages()
    void reculer(); // décrémente _posImageCourante, modulo nblImages()
    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 nblImages(); // affiche la taille de _diaporama
    Image* imageCourante(); // retourne le pointeur vers l'image courante
    unsigned int numDiaporamaCourant();

private:
    unsigned _numDiaporamaCourant; // numéro du diaporama courant, par défaut 0
    Diaporama _diaporama; // pointeurs vers les images du diaporama
    unsigned int _posImageCourante; /* position, dans le diaporama,
                                     de l'image courante.
                                     Indéfini quand diaporama vide.
                                     Démarre à 0 quand diaporama non vide */

private:
    void chargerDiaporama(); // charge dans _diaporama les images du _numDiaporamaCourant
    void viderDiaporama(); // vide _diaporama de tous ses objets image et les delete
};

#endif // LECTEUR_H
```

Figure 4 : Schéma de classes = Classe XXX

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

(d) Remarques concernant le schéma de classes

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

4. Implémentation et tests

4.1 Implémentation

Liste et rôle des fichiers de cette version :

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

4.2 Test

Test avec le programme fournit main.cpp

Valeurs attendus

```
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
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
Press <RETURN> to close this window...
```

Valeurs réelle

```

vide
Diaporama num. 1 selectionne.
4 images chargees dans le diaporama
1
image( rang:1, titre:Grincheux, categorie:personne, chemin:C:\cartesDisney\carteDisney1.gif)
image( rang:2, titre:Cendrillon, categorie:personne, chemin:C:\cartesDisney\carteDisney4.gif)
image( rang:3, titre:Blanche Neige, categorie:personne, chemin:C:\cartesDisney\carteDisney2.gif)
image( rang:4, titre:Mickey, categorie:animal, chemin:C:\cartesDisney\carteDisney1.gif)

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

```

```

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

```


Version v1 – projet Graphique seul

5. Éléments d'interface

<i>Nom variable manipulée par l'application</i>	<i>Nom sur l'interface</i>
INomDiapo	Nom du diaporama
IRang	valRang
ITitre	Titre
ICategorie	valCategorie
bImage	rien d'écrit car contiendra plus tard l'image
bPrecedent	<-
bSuivant	->
bLancerDiapo	Lancer Diaporama
bArreterDiapo	Arrêter Diaporama
actionQuitter	(Fichier->) Quitter
actionCharger_diaporama	(Paramètres->) Charger Diaporama
actionEnlever_diaporama	(Paramètres->) Enlever Diaporama
actionVitesse_de_defilement	(Paramètres->) Vitesse de défilement
actionA_propos_de	(Aide->) A propos de...

6. Implémentation et tests

6.1 Implémentation

Liste et rôle des fichiers de cette version :

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

Remarques sur l'implémentation :

- Lors d'un click (signal : clicked) sur le bouton "Lancer Diaporama" l'événement (slot) LancerDiapo() est lancé (cela permettra plus tard le lancement en mode automatique du diaporama ainsi que l'affichage du mode dans la status bar).
- Lors d'un click (signal : clicked) sur le bouton "Arrêter Diaporama" l'événement (slot) ArreterDiapo() est

lancé (cela permettra plus tard le l'arrêt en mode automatique du diaporama et le lancement en mode manuel ainsi que l'affichage du mode dans la status bar).

- Lors d'un clic (signal : clicked) sur le bouton "->" l'événement (slot) AllerAuSuivant() est lancé (cela permettra plus tard de passer à la diapositive suivante en mode manuel).
- Lors d'un clic (signal : clicked) sur le bouton "<-" l'événement (slot) AllerAuPrecedent() est lancé (cela permettra plus tard de passer à la diapositive précédente en mode manuel).
- Lors d'un clic (signal : triggered) sur l'item de menu "Charger diaporama" l'événement (slot) ChargerDiapo() est lancé (cela permettra plus tard de charger le diaporama).
- Lors d'un clic (signal : triggered) sur l'item de menu "Enlever diaporama" l'événement (slot) EnleverDiapo() est lancé (cela permettra plus tard d'enlever le diaporama).
- Lors d'un clic (signal : triggered) sur l'item de menu "Vitesse de défilement" l'événement (slot) ChangerVitesse() est lancé (cela permettra plus tard de modifier la vitesse de défilement entre les diapositives du diaporama).
- Lors d'un clic (signal : triggered) sur l'item de menu "A propos de..." l'événement (slot) SeRenseigner() est lancé (cela permettra plus tard d'afficher la version, la date de création et les auteurs de l'application).
- Lors d'un clic sur l'item de menu "Quitter" la fenêtre de l'application se ferme.

6.2 Test

Tests prévus :

- Clic sur un bouton ou un item de menu affiché sur l'interface ("Lancer diaporama", "Arrêter diaporama", "->", "<-", "Charger diaporama", "Enlever diaporama", "Vitesse de défilement", "A propos de...", "Quitter").

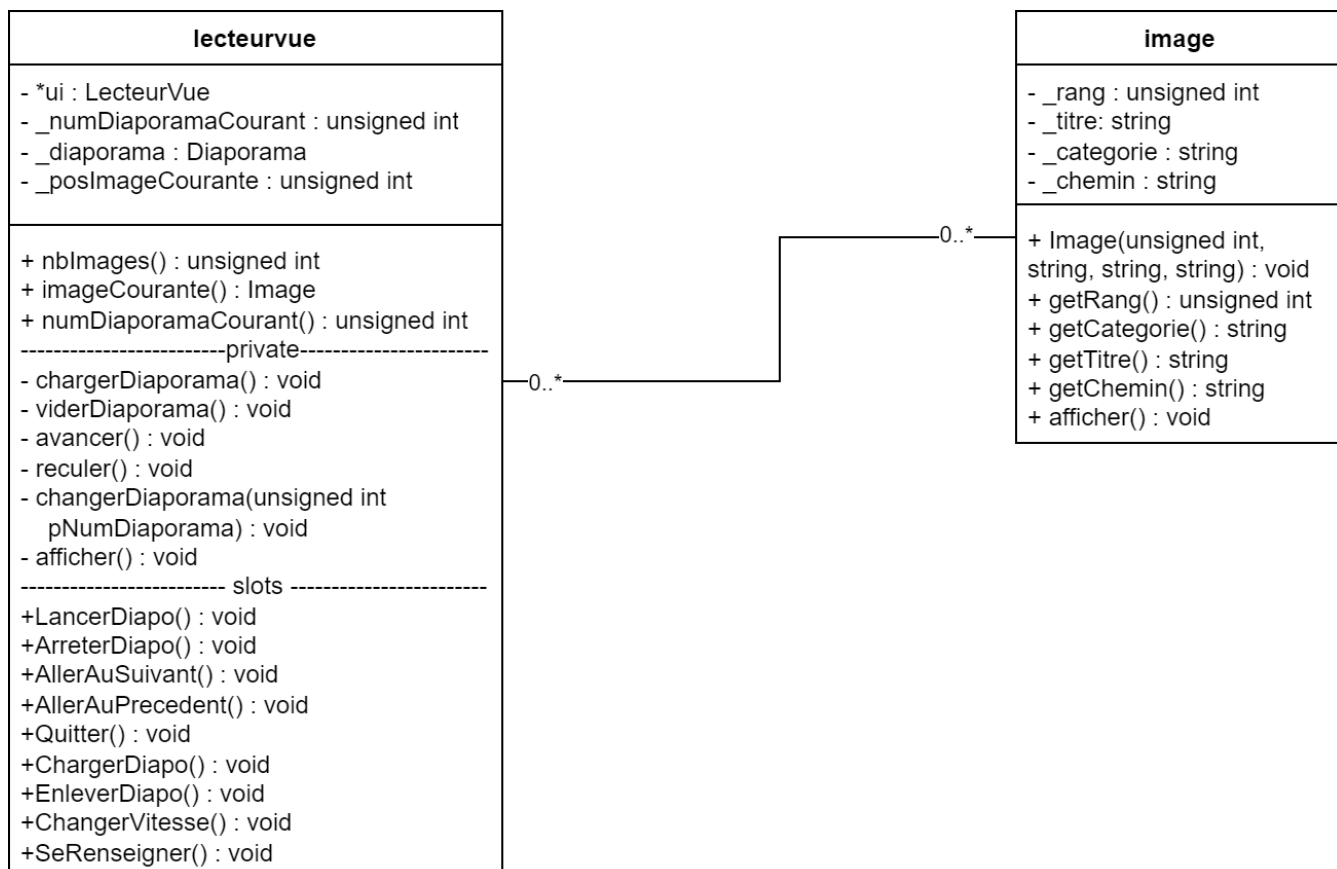
Tests réalisés :

- Lors d'un clic sur un bouton ou un item de menu affiché sur l'interface un message (Qdebug()) s'affiche dans la console pour confirmer l'exécution du clic donc le lancement de l'évènement (slot) lié.

Nous n'avons pas de comportement lié aux aspects non-fonctionnels du programme car cela pourrait concerner les performances du programmes qui ne sont pas testables à cette version.

Diagramme de classes (UML)

(a) Dictionnaire des éléments pour chaque classe



(b) Dictionnaire des éléments pour chaque classe

Inchangé hormis le nom lecteur qui devient lecteurvue

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

Classe lecteurVue :

```

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

```

QT_BEGIN_NAMESPACE
namespace Ui { class LecteurVue; }
  
```

QT_END_NAMESPACE

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

class LecteurVue : public QMainWindow

{

Q_OBJECT

public:

LecteurVue(QWidget *parent = nullptr);

~LecteurVue();

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

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

unsigned int numDiaporamaCourant();

private:

Ui::LecteurVue *ui;

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

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

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

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

public slots:

void LancerDiapo();

void ArreterDiapo();

void AllerAuSuivant();

void AllerAuPrecedent();

```

void Quitter();
void ChargerDiapo();
void EnleverDiapo();
void ChangerVitesse();
void SeRenseigner();
};
#endif // LECTEURVUE_H

```

Les autres classes ne sont pas modifiées.

7. Comportement de l’application

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

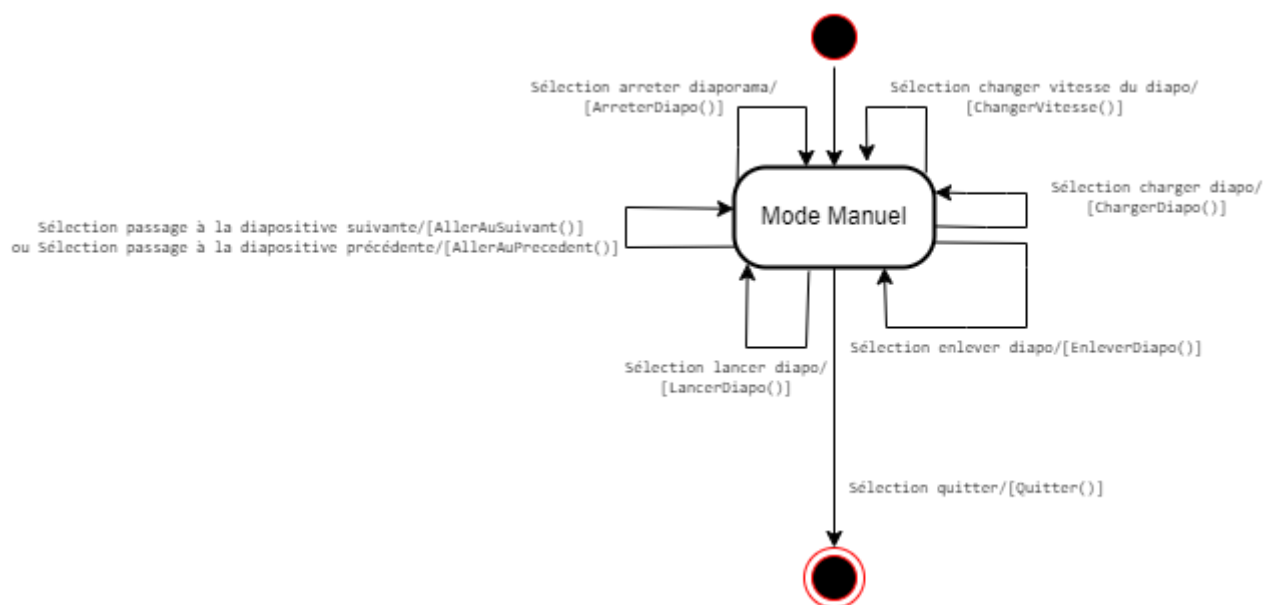


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v2

7.2 Dictionnaire des états, événements et Actions (v2)

Dictionnaire des états du diaporama	
nomEtat	Signification
Mode Manuel	Le lecteur de diaporama à un diaporama chargé ou nom. Lors de ce mode les diapositives du diaporama ne défilent pas automatiquement mais à l’aide des flèches (en tant que boutons) mises à disposition.

Tableau 2 : États du lecteur de diaporamas – v2

Dictionnaire des événements faisant changer le diaporama d’état

nomÉvénement	Signification
Sélection charger diapo	Clic sur l’item de menu “Charger diaporama”
Sélection enlever diapo	Clic sur l’item de menu “Enlever diaporama”
Sélection lancer diapo	Clic sur le bouton “Lancer diaporama”
Sélection arrêter diapo	Clic sur le bouton “Arrêter diaporama”
Sélection passage à la diapositive suivante	Clic sur le bouton “->”
Sélection passage à la diapositive précédente	Clic sur le bouton “<-”
Sélection changer vitesse diapo	Clic sur l’item de menu “Vitesse de défilement”
Sélection quitter diapo	Clic sur l’item de menu “Quitter”

Tableau 3 : Événements faisant changer le diaporama d’état – v2

Description des actions réalisées lors de la traversée des transitions

nomAction	Signification
ChargerDiapo()	Lors de cette version cela charge les information de la diapo stocké en dur
EnleverDiapo()	Lors de cette version cela affiche un message qDebug() dans la console pour valider le lancement de l’évènement
LancerDiapo()	Lors de cette version cela affiche un message qDebug() dans la console pour valider le lancement de l’évènement
ArreterDiapo()	Lors de cette version cela affiche un message qDebug() dans la console pour valider le lancement de l’évènement
AllerAuSuivant()	Passage d’une diapositive à la diapositive suivante.
AllerAuPrecedent()	Passage d’une diapositive à la diapositive précédente.
ChangerVitesse()	Lors de cette version cela affiche un message qDebug() dans la console pour valider le lancement de l’évènement
Quitter()	Fermeture de la fenêtre du lecteur de diaporama et arrêt de l’application

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

7.3 Table T_EtatsEvenementsActions (v2)

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

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

Élément graphique prenant en charge cet événement ☒	item menu "Charger Diaporama"	item menu "Enlever Diaporama"	Bouton "Lancer Diaporama"	Bouton "Arrêter Diaporama"	Bouton "->"	Bouton "<-"	item menu "Vitesse de defilement"	item menu "Quitter"
Événement ☒ nomEtat	Sélection charger diapo	Sélection enlever diapo	Sélection lancer diapo	Sélection arrêter diapo	Sélection passage à la diapositive suivante	Sélection passage à la diapositive précédente	Sélection changer vitesse diapo	Sélection quitter diapo
Mode manuel	Aucun Etat d'arrivée de transition / ChargerDiapo()	Aucun Etat d'arrivée de transition / EnleverDiapo()	Aucun Etat d'arrivée de transition / LancerDiapo()	Aucun Etat d'arrivée de transition / pas d'action faisant changer l'état	Aucun Etat d'arrivée de transition / AllerAuSuivant()	Aucun Etat d'arrivée de transition / AllerAuPrecedent()	Aucun Etat d'arrivée de transition / pas d'action faisant changer l'état	Aucun Etat d'arrivée de transition / Quitter()

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

8. Implémentation et tests

8.1 Implémentation (v2)

Liste et rôle des fichiers de cette version :

lecteurVue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas
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.
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image
image.cpp	Corps de la classe Image
main.cpp	Programme principal où l'on lance le fonctionnement de l'application

Remarques sur l'implémentation :

Lors de cette version les slots liés aux passage d'une image à une autre ainsi que la possibilité de quitter la fenêtre en cliquant sur "quitter" dans les paramètres ou encore d'obtenir des renseignements sur le lecteur de diaporama en cliquant sur "A propos de..." dans le menu aide deviennent fonctionnels. Les autres se contentent de communiquer comme lors de la version précédente.

8.2 Tests (v2)

Tests prévus :

- vérifier si le rang de l'image s'affiche ;
- vérifier si la catégorie de l'image s'affiche ;

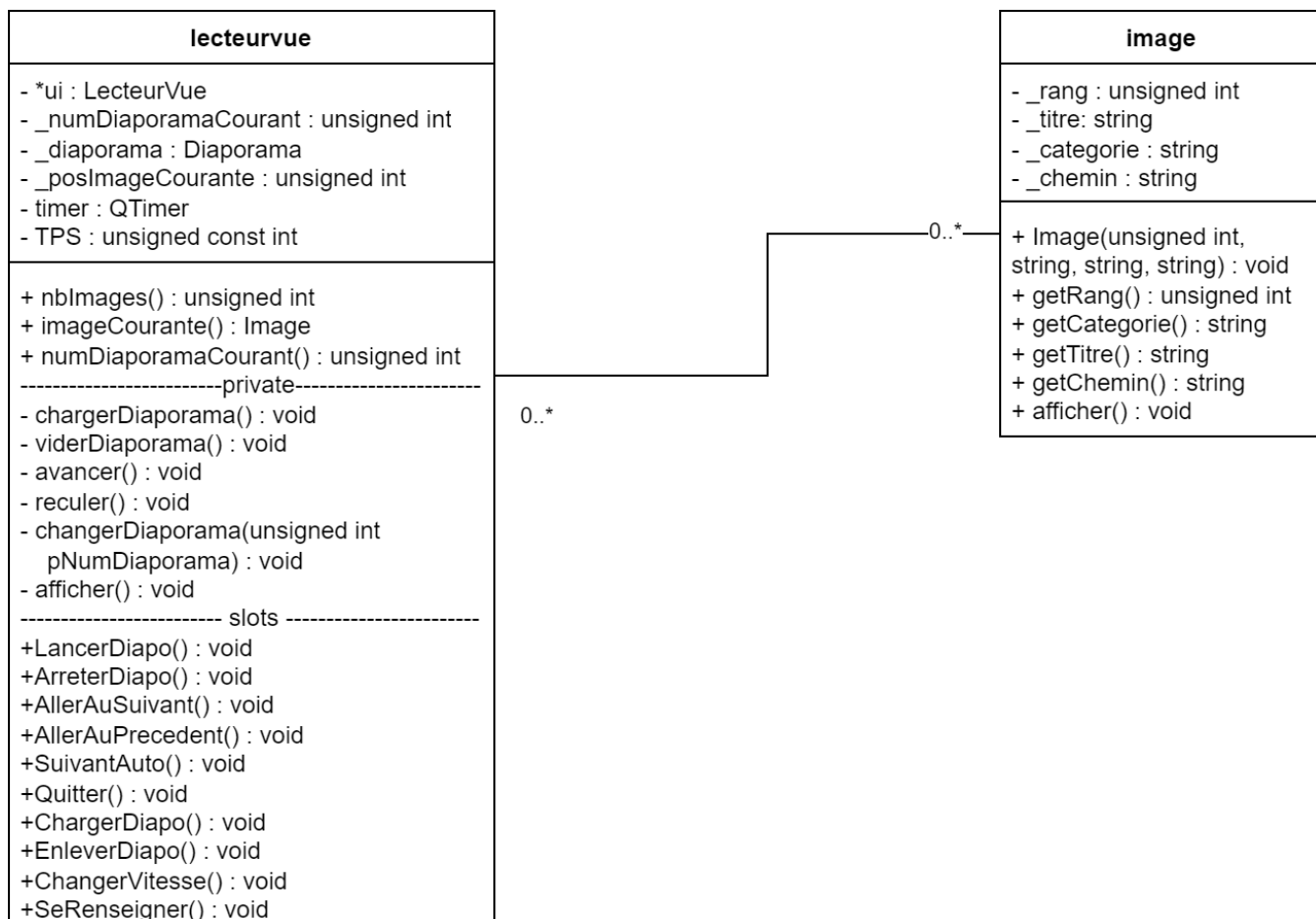
- vérifier si le titre de l'image s'affiche ;
- clic sur l'item de menu "quitter" ;
- clic sur l'item de menu "A propos de..." ;
- clic sur le bouton "->" ;
- clic sur le bouton "<-" ;
- aller à la diapo suivante ou précédente en fin ou début de diaporama .

Tests réalisés :

- le rang de l'image s'affiche ;
- la catégorie de l'image s'affiche ;
- le titre de l'image s'affiche ;
- Lors d'un clic sur l'item de menu "quitter" la fenêtre se ferme et l'application s'arrête ;
- Lors d'un clic sur l'item de menu "A propos de..." la fenêtre les renseignements concernant la version de l'application s'affichent ;
- Lors d'un clic sur le bouton "->" le lecteur de diaporama passe à l'image suivante ;
- Lors d'un clic sur le bouton "<-" le lecteur de diaporama passe à l'image précédente ;
- lors de l'action d'aller à la diapo suivante ou précédente en fin ou début de diaporama, on passe à la première diapositive ou la dernière suivant le cas ;
- Les images s'affichent comme souhaité dans le lecteur de diaporama.

9. Diagramme de classes (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 lecteurvue			
Nom attribut	Signification	Type	Exemple
_numDiaporamaCourant	Le numéro courant du diaporama manipulé	unsigned int	1
_diaporama	Quelle diaporama est en cours de manipulation	vector<*Image>Diaporama	"mes vacances 2021"
_posImageCourante	La position de l'image courante	unsigned int	3
timer	Système de minuteries répétitives et uniques qui émet un signal à chaque intervalle de temps donné en milliseconde	QTimer	2000
TPS	La durée du timer	unsigned const int	2000

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

```
#ifndef LECTEURVUE_H
#define LECTEURVUE_H
#include "image.h"
#include <vector>
#include <QMainWindow>
#include<QTimer>

QT_BEGIN_NAMESPACE
namespace Ui { class LecteurVue; }
QT_END_NAMESPACE

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

class LecteurVue : public QMainWindow
{
    Q_OBJECT

public:
    LecteurVue(QWidget *parent = nullptr);
    ~LecteurVue();
    unsigned int nbImages(); // affiche la taille de _diaporama
    Image* imageCourante(); // retourne le pointeur vers l'image courante
    unsigned int numDiaporamaCourant();

private:
    Ui::LecteurVue *ui;
    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 */
    QTimer timer;
    unsigned const int TPS = 2000;
private:
    void chargerDiaporama(); // charge dans _diaporama les images du _numDiaporamaCourant
    void viderDiaporama(); // vide _diaporama de tous ses objets image et les delete
    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

public slots:
    void LancerDiapo();
    void ArreterDiapo();
    void AllerAuSuivant();
    void AllerAuPrecedent();
    void SuivantAuto();
    void Quitter();
    void ChargerDiapo();
```

```

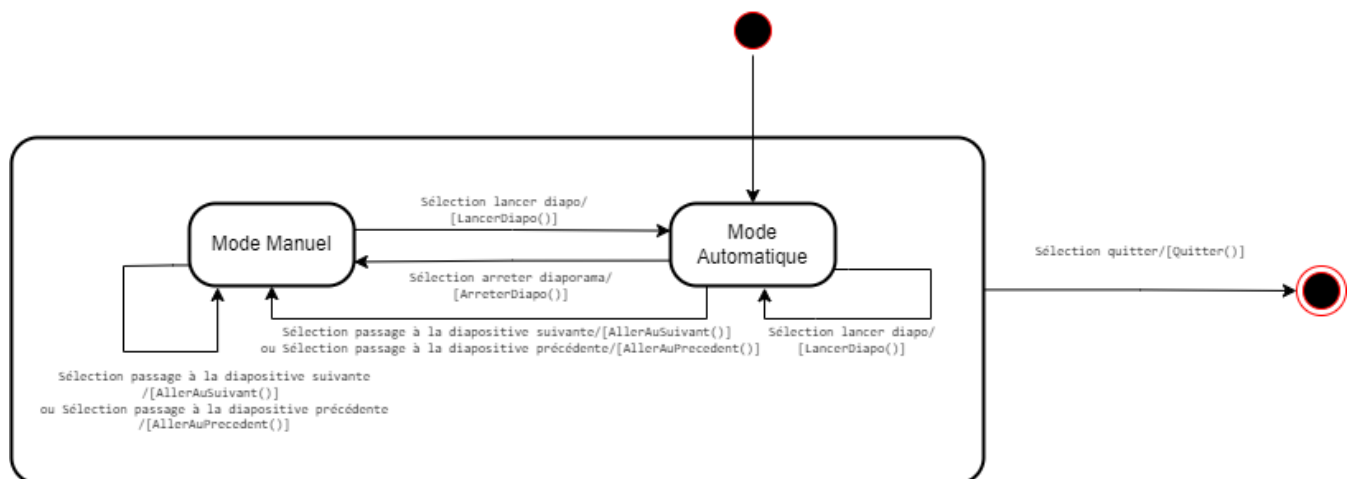
void EnleverDiapo();
void ChangerVitesse();
void SeRenseigner();
};
#endif // LECTEURVUE_H

```

Les autres classes ne sont pas modifiées.

10. Comportement de l'application

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



7.2 Dictionnaire des états, événements et Actions (v3)

Dictionnaire des états du diaporama

nomEtat	Signification
Mode Manuel	Le lecteur de diaporama à un diaporama chargé. Lors de ce mode les diapositives du diaporama ne défilent pas automatiquement mais à l'aide des flèches (en tant que boutons) mises à disposition.
Mode Automatique	Le lecteur de diaporama à un diaporama chargé. Lors de ce mode les diapositives défilent automatiquement avec une vitesse prédéfinie qui sera modifiable dans les prochaines versions.

Tableau 2 : États du lecteur de diaporamas – v3

Dictionnaire des événements faisant changer le diaporama d'état

nomEvénement	Signification
Sélection charger diapo	Clic sur l'item de menu "Charger diaporama"
Sélection enlever diapo	Clic sur l'item de menu "Enlever diaporama"
Sélection lancer diapo	Clic sur le bouton "Lancer diaporama"
Sélection arrêter diapo	Clic sur le bouton "Arrêter diaporama"
Sélection passage à la diapositive suivante	Clic sur le bouton "->"

Sélection passage à la diapositive précédente	Clic sur le bouton "<-"
Sélection changer vitesse diapo	Clic sur l'item de menu "Vitesse de défilement"
Sélection quitter diapo	Clic sur l'item de menu "Quitter"
Fin du timer après la sélection du mode auto	Clic sur le bouton "Lancer diaporama"

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

Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>
ChargerDiapo()	Lors de cette version cela affiche un message qdebug() dans la console pour valider le lancement de l'évènement
EnleverDiapo()	Lors de cette version cela affiche un message qdebug() dans la console pour valider le lancement de l'évènement
LancerDiapo()	Passage en mode automatique du défilement des diapositives, c'est-à-dire que les images défilent à une certaine vitesse prédéfinie. Aussi lorsque nous sommes déjà en mode automatique nous réinitialiser simplement le diaporama en restant en mode auto et en recommençant à la première diapositive.
ArreterDiapo()	Passage en mode manuel lorsque le mode automatique est activé, c'est-à-dire qu'il faudra ensuite manuellement passer d'une image à l'autre.
AllerAuSuivant()	Passage d'une diapositive à la diapositive suivant lorsque le mode manuel est activé. Si l'on clique dessus lorsque le mode auto est activé, la lecture du diaporama passe en mode manuel.
AllerAuPrecedent()	Passage d'une diapositive à la diapositive précédente lorsque le mode manuel est activé. Si l'on clique dessus lorsque le mode auto est activé, la lecture du diaporama passe en mode manuel.
ChangerVitesse()	Lors de cette version cela affiche un message qdebug() dans la console pour valider le lancement de l'évènement
Quitter()	Fermeture de la fenêtre du lecteur de diaporama et arrêt de l'application
SuivantAuto()	Permet de passer automatiquement d'une image à l'autre grâce à l'instanciation d'un timer qui dès qu'il se finit rappelle ce slot pour passer à l'image suivante et l'afficher.

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

7.3 Table T_EtatsEvenementsActions (v3)

Élément graphique prenant en charge cet événement ☐	item menu "Charger Diaporama"	item menu "Enlever Diaporama"	Bouton "Lancer Diaporama"	Bouton "Arrêter Diaporama"	Bouton "->"	Bouton "<-"	item menu "Vitesse de défilement"	item menu "Quitter"
Événement ☐ nomEtat	Sélection charger diapo	Sélection enlever diapo	Sélection lancer diapo	Sélection arrêter diapo	Sélection passage à la diapositive suivante	Sélection passage à la diapositive précédente	Sélection changer vitesse diapo	Sélection quitter diapo
Mode manuel	Mode Attente / pas d'action faisant changer l'état	Mode Attente / pas d'action faisant changer l'état	Mode Attente / LancerDiapo()	Mode Attente / pas d'action faisant changer l'état	Mode Attente / AllerAuSuivant()	Mode Attente / AllerAuPrecedent()	Mode Attente / pas d'action faisant changer l'état	Mode Attente / Quitter()
Mode automatique	Mode Manuel / pas d'action faisant changer l'état	Mode Manuel / pas d'action faisant changer l'état	Mode Manuel / pas d'action faisant changer l'état	Mode Manuel / ArrêterDiapo()	Mode Manuel / AllerAuSuivant()	Mode Manuel / AllerAuPrecedent()	Mode Manuel / pas d'action faisant changer l'état	Mode Manuel / Quitter()

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

11. Implémentation et tests

8.1 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
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.
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image
image.cpp	Corps de la classe Image
main.cpp	Programme principal où l'on lance le fonctionnement de l'application

Remarques sur l'implémentation :

Lors de cette version les slots liés au lancement du diaporama donc au passage automatique du défilement des images, et à l'arrêt du diaporama sont fonctionnels. Les autres se contentent de communiquer ou de faire les actions qui leurs sont attribuées comme lors de la version précédente.

8.2 Tests (v3)

Tests prévus :

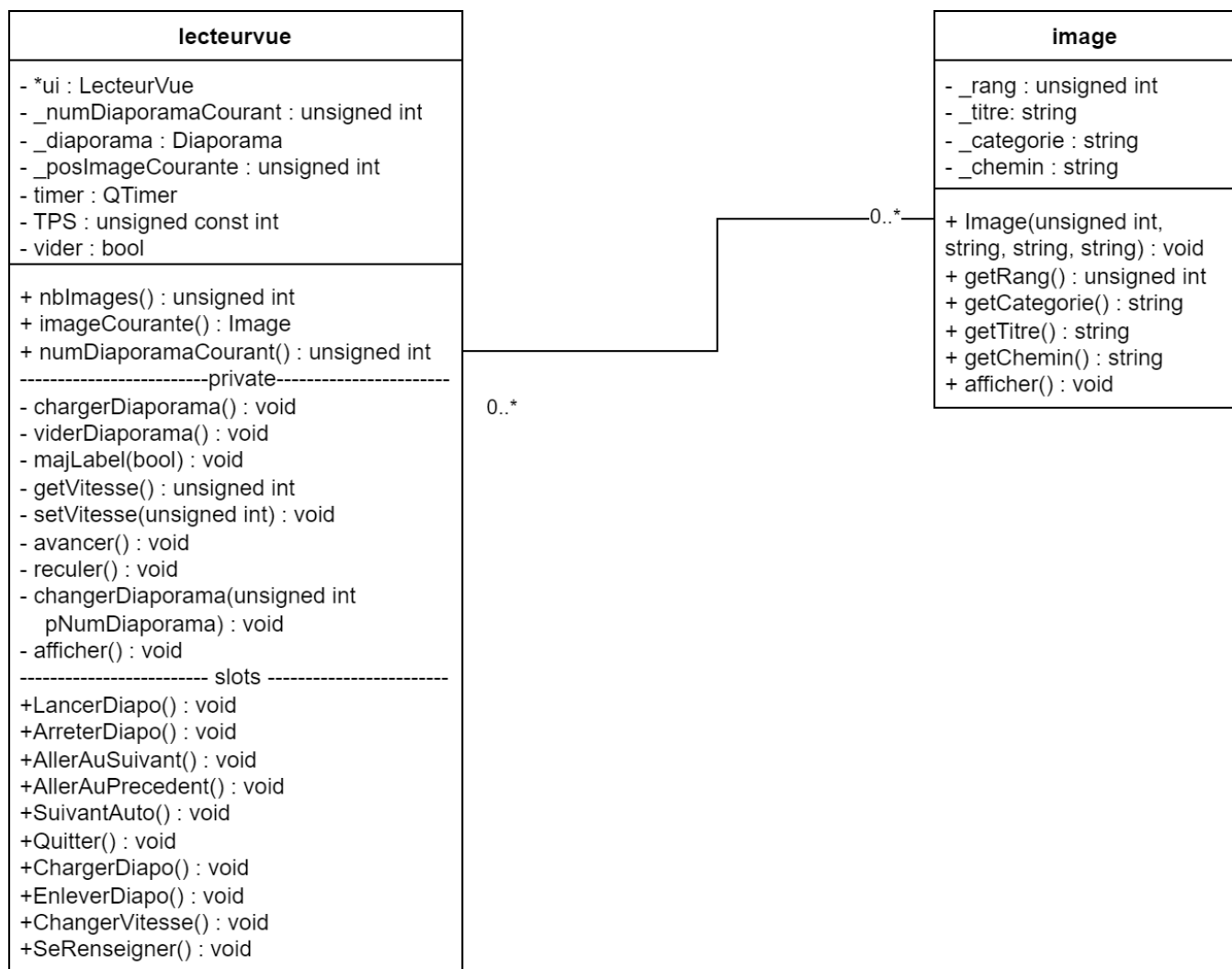
- clic sur le bouton "Lancer diaporama" ;

- clic sur le bouton “Arreter diaporama” ;
- aller à la diapo suivante ou précédente lorsque le mode auto est activé ;
- lancer diaporama lorsque le mode auto est activé.

Tests réalisés :

- lors du clic sur le bouton “Lancer diaporama” les images du diaporama défilent de manière automatique avec une vitesse prédéfinie ;
- lors du clic sur le bouton “Arreter diaporama” les images du diaporama arrêtent de défiler et le mode automatique est arrêté pour passer au mode manuel ;
- lorsque le mode auto est activé et que l’on essaye d’aller à la diapo suivante ou précédente, arrête le diaporama et passe en mode manuel ;
- lors du lancement du diaporama lorsque le mode auto est activé, on relance le diaporama chargé à la première diapositive et on reste en mode auto.

12. Diagramme de classes (UML)



(b) Dictionnaire des éléments pour chaque classe

Classe lecteurvue			
Nom attribut	Signification	Type	Exemple
_numDiaporamaCourant	Le numéro courant du diaporama manipulé	unsigned int	1
_diaporama	Quelle diaporama est en cours de manipulation	vector<*Image>Diaporama	"mes vacances 2021"
_posImageCourante	La position de l'image courante	unsigned int	3
timer	Système de minuteries répétitives et uniques qui émet un signal à chaque intervalle de temps donné en milliseconde	QTimer	2000

TPS	La durée du timer	unsigned const int	2000
vider	savoir si on doit afficher ou non les éléments graphique	bool	true

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

```
#ifndef LECTEURVUE_H
#define LECTEURVUE_H
#include "image.h"
#include <vector>
#include <QMainWindow>
#include<QTimer>
```

```
QT_BEGIN_NAMESPACE
namespace Ui { class LecteurVue; }
QT_END_NAMESPACE
```

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

```
class LecteurVue : public QMainWindow
{
    Q_OBJECT
```

public:

```
LecteurVue(QWidget *parent = nullptr);
~LecteurVue();
unsigned int nbImages(); // affiche la taille de _diaporama
Image* imageCourante(); // retourne le pointeur vers l'image courante
unsigned int numDiaporamaCourant();
```

private:

```
Ui::LecteurVue *ui;
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 */
```

```
QTimer timer;
unsigned int TPS = 2000; // durer du timer
bool vider = false; // indicateur pour afficher les textes
```

private:

```
void chargerDiaporama(); // charge dans _diaporama les images du _numDiaporamaCourant
void viderDiaporama(); // vide _diaporama de tous ses objets image et les delete
void majLabel(bool);
unsigned int getVitesse(); // retourne la vitesse à laquelle défile les diapositives
void setVitesse(unsigned int); // change la vitesse à laquelle défile les diapositives selon l'entier donné
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
```

```
public slots:
```

```
void LancerDiapo();
void ArreterDiapo();
void AllerAuSuivant();
void AllerAuPrecedent();
void SuivantAuto();
void Quitter();
void ChargerDiapo();
void EnleverDiapo();
void ChangerVitesse();
void SeRenseigner();
```

```
};
```

```
#endif // LECTEURVUE_H
```

Les autres ne sont pas modifier

13. Comportement de l'application

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

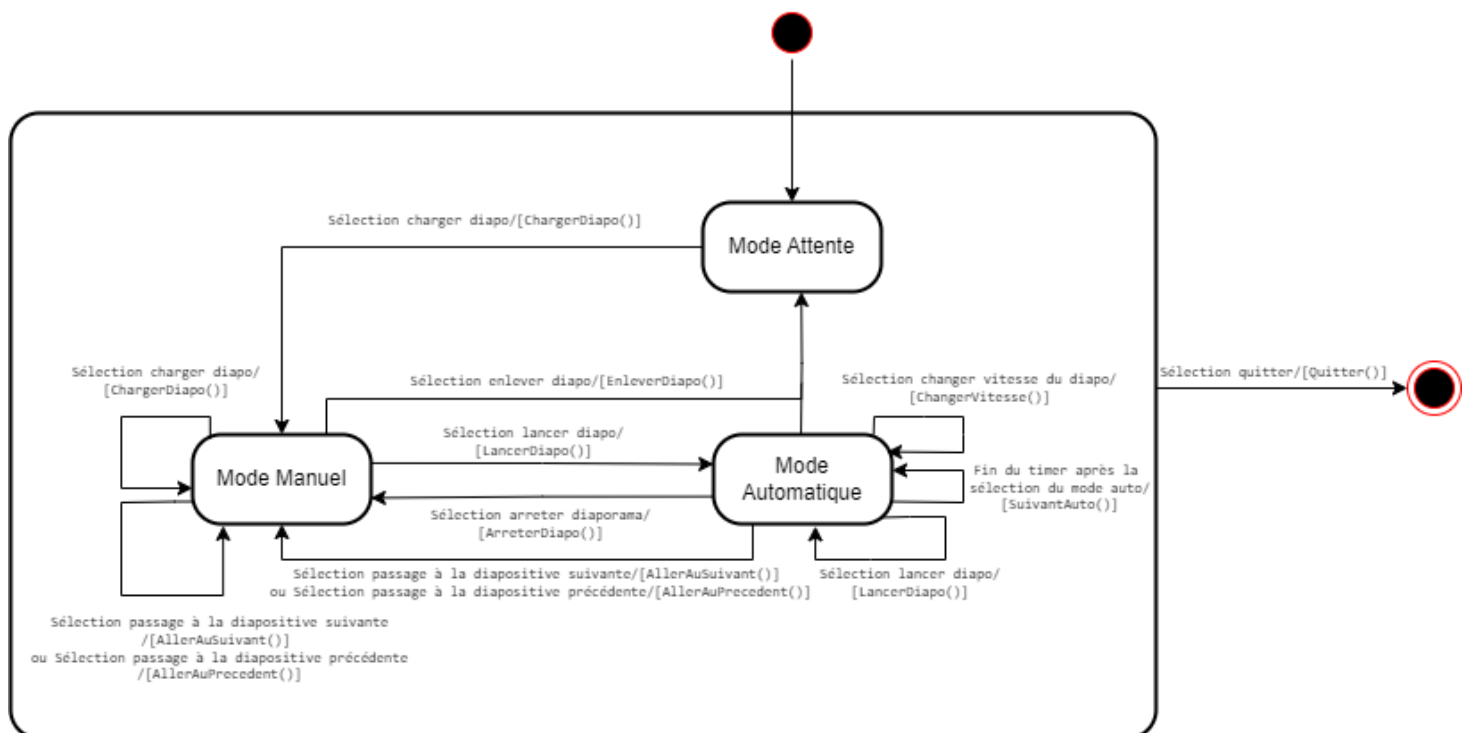


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v2

7.2 Dictionnaire des états, événements et Actions (v4)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
Mode Attente	Le lecteur de diaporama n'a aucun diaporama chargé, il attend donc le chargement d'un diaporama pour ensuite passer en mode manuel.
Mode Manuel	Le lecteur de diaporama à un diaporama chargé. Lors de ce mode les diapositives du diaporama ne défilent pas automatiquement mais à l'aide des flèches (en tant que boutons) mises à disposition.
Mode Automatique	Le lecteur de diaporama à un diaporama chargé. Lors de ce mode les diapositives défilent automatiquement avec une vitesse prédéfinie qui est modifiable.

Tableau 2 : États du lecteur de diaporamas – v4

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomEvénement</i>	<i>Signification</i>
Sélection charger diapo	Clic sur l'item de menu "Charger diaporama"
Sélection enlever diapo	Clic sur l'item de menu "Enlever diaporama"
Sélection lancer diapo	Clic sur le bouton "Lancer diaporama"
Sélection arrêter diapo	Clic sur le bouton "Arrêter diaporama"
Sélection passage à la diapositive suivante	Clic sur le bouton ">"
Sélection passage à la diapositive précédente	Clic sur le bouton "<"
Sélection changer vitesse diapo	Clic sur l'item de menu "Vitesse de défilement"
Sélection quitter diapo	Clic sur l'item de menu "Quitter"
Fin du timer après la sélection du mode auto	Clic sur le bouton "Lancer diaporama"

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

Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>
ChargerDiapo()	Lors de cette version, le chargement de diaporama permet d'aller récupérer les images qui sont stockées en dur et dont le chemin ainsi que les informations associées sont inscrit dans le code.
EnleverDiapo()	Lors de cette version enlever un diaporama correspond à ne plus aller récupérer les images du diaporama qui sont stockés en dur ainsi que lire leurs informations
LancerDiapo()	Passage en mode automatique du défilement des diapositives, c'est-à-dire que les images défilent à une certaine vitesse prédéfinie. Aussi lorsque nous sommes déjà en mode automatique nous réinitialiser simplement le diaporama en restant en mode auto et en recommençant à la première diapositive.

ArreterDiapo()	Passage en mode manuel lorsque le mode automatique est activé, c'est-à-dire qu'il faudra ensuite manuellement passer d'une image à l'autre.
AllerAuSuivant()	Passage d'une diapositive à la diapositive suivant lorsque le mode manuel est activé. Si l'on clique dessus lorsque le mode auto est activé, la lecture du diaporama passe en mode manuel.
AllerAuPrecedent()	Passage d'une diapositive à la diapositive précédente lorsque le mode manuel est activé. Si l'on clique dessus lorsque le mode auto est activé, la lecture du diaporama passe en mode manuel.
ChangerVitesse()	Lors de cette version cela affiche un message qdebug() dans la console pour valider le lancement de l'évènement
Quitter()	Fermeture de la fenêtre du lecteur de diaporama et arrêt de l'application
SuivantAuto()	Permet de passer automatiquement d'une image à l'autre grâce à l'instanciation d'un timer qui dès qu'il se finit rappelle ce slot pour passer à l'image suivante et l'afficher.

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

7.3 Table T_EtatsEvenementsActions (v4)

Élément graphique prenant en charge cet événement ☐	item menu "Charger Diaporama"	item menu "Enlever Diaporama"	Bouton "Lancer Diaporama"	Bouton "Arrêter Diaporama"	Bouton "->"	Bouton "<-"	item menu "Vitesse de défilement"	item menu "Quitter"	Bouton "Lancer Diaporama"
Événement ☐ nomEtat	Sélection charger diapo	Sélection enlever diapo	Sélection lancer diapo	Sélection arrêter diapo	Sélection passage à la diapositive suivante	Sélection passage à la diapositive précédente	Sélection changer vitesse diapo	Sélection quitter diapo	Fin du timer après la sélection du mode auto
Mode attente	Pas d'état à l'arrivée / ChargerDiapo()	Pas d'état à l'arrivée / pas d'action faisant changer l'état	Pas d'état à l'arrivée / pas d'action faisant changer l'état	Pas d'état à l'arrivée / pas d'action faisant changer l'état	Pas d'état à l'arrivée / pas d'action faisant changer l'état	Pas d'état à l'arrivée / pas d'action faisant changer l'état	Pas d'état à l'arrivée / pas d'action faisant changer l'état	Pas d'état à l'arrivée / Quitter()	Pas d'état à l'arrivée / pas d'action faisant changer l'état
Mode manuel	Mode Attente / ChargerDiapo()	Mode Attente / EnleverDiapo()	Mode Attente / LancerDiapo()	Mode Attente / pas d'action faisant changer l'état	Mode Attente / AllerAuSuivant()	Mode Attente / AllerAuPrécédent()	Mode Attente / pas d'action faisant changer l'état	Mode Attente / Quitter()	Mode Attente / pas d'action faisant changer l'état
Mode automatique	Mode Manuel / pas d'action faisant changer l'état	Mode Manuel / EnleverDiapo()	Mode Manuel / pas d'action faisant changer l'état	Mode Manuel / ArrêterDiapo()	Mode Manuel / AllerAuSuivant()	Mode Manuel / AllerAuPrécédent()	Mode Manuel / ChangerVitesse()	Mode Manuel / Quitter()	Mode Manuel / SuivantAuto()

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

14. Implémentation et tests

8.1 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
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.
lecteur.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image
image.cpp	Corps de la classe Image
main.cpp	Programme principal où l'on lance le fonctionnement de l'application

Remarques sur l'implémentation :

Lors de cette version les slots liés au changement de vitesse de défilement des images, au chargement de diaporama ou à l'enlèvement de diaporama sont fonctionnels. Les autres se contentent de faire les actions qui leurs sont attribuées comme lors de la version précédente. Aussi nous avons ajouté le slot suivant `Auto()` qui permet de passer automatiquement d'une image à l'autre en prenant compte de l'intervalle de temps entre chaque image.

8.2 Tests (v4)

Tests prévus :

- clic sur l'item de menu "Charger un diaporama" ;
- clic sur l'item de menu "Enlever un diaporama" ;
- sélectionner autre chose qu'un nombre ou un nombre négatif dans le changement de vitesse du diaporama.
- arrêter, lancer le diaporama, aller à la diapo suivante ou précédente lorsque le diaporama n'est pas chargé ;
- charger un diaporama lorsqu'un diaporama est déjà chargé ou enlever un diaporama lorsqu'aucun diaporama n'est chargé.

Tests réalisés :

- lors clic sur l'item de menu "Charger un diaporama" un diaporama que nous avons préalablement choisi dans le code (chargé en dur) est chargé et la première image du diaporama ainsi que les informations associées s'affichent ;
- lors clic sur l'item de menu "Enlever un diaporama" les informations des images ainsi que les images d'un diaporama que nous avons préalablement choisi dans le code sont retirées de l'affichage ;
- lors du changement de vitesse du diaporama il n'est possible de valider que des nombres compris entre 0 et 120 et il est impossible de saisir des lettres (car utilisation d'une zone de saisie qui permet de saisir seulement des valeurs possibles entre 0 et 120 (pour 120 sec donc 2 min)) ;
- lors de l'arrêt ou du lancement du diaporama ou lors de l'action d'aller à la diapo précédente ou suivante lorsque le diaporama n'est pas chargé, aucuns des boutons ne sont cliquables ;
- lors du chargement d'un diaporama lorsqu'un diaporama n'est chargé ou lors de l'enlèvement d'un diaporama lorsqu'aucun diaporama n'est chargé, un message d'erreur s'affiche.