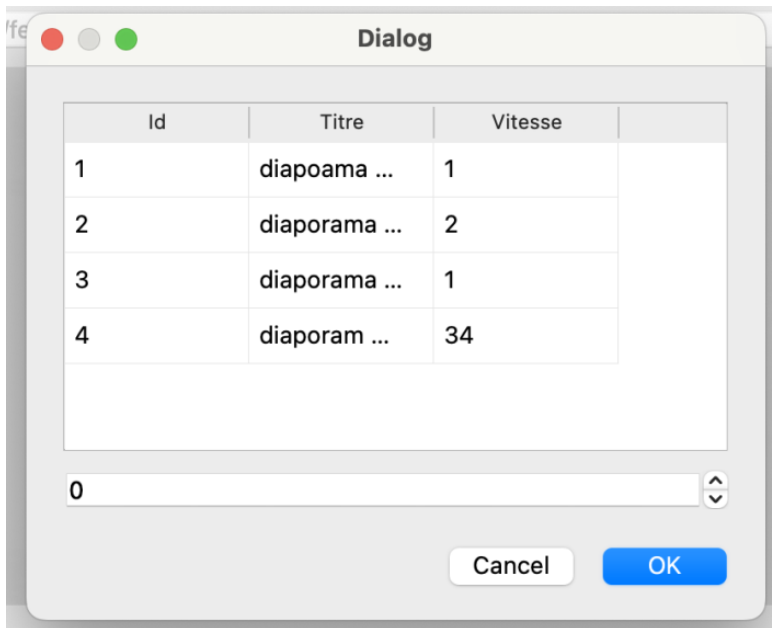


Saé 2.01 – Développement d'une application

Lecteur de diaporamas – Dossier d'Analyse et conception

1. Compléments de spécifications externes



Fenêtre "Choix Diaporama"

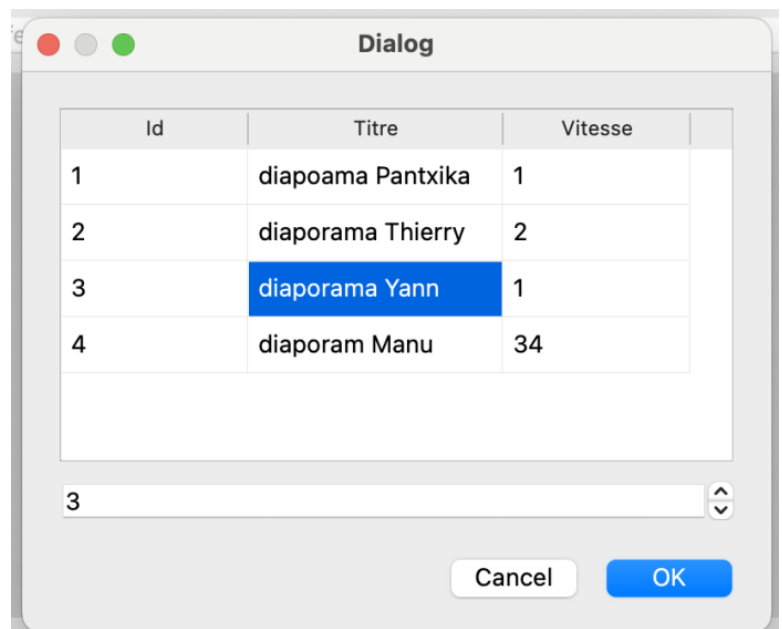
L'utilisateur choisit le diaporama parmi ceux proposés en cliquant dessus ou en saisissant l'identifiant du diaporama

Cette fenêtre s'affiche après avoir cliquer sur le menu "Charger Diaporama"

Fenêtre "Choix Diaporama cliquée"

L'utilisateur peut cliquer sur n'importe quelle colonne de la ligne du diaporama que l'on veut sélectionner

Le système indique en mettant en surbrillance la case sélectionner et change l'identifiant dans le champ de text en bas

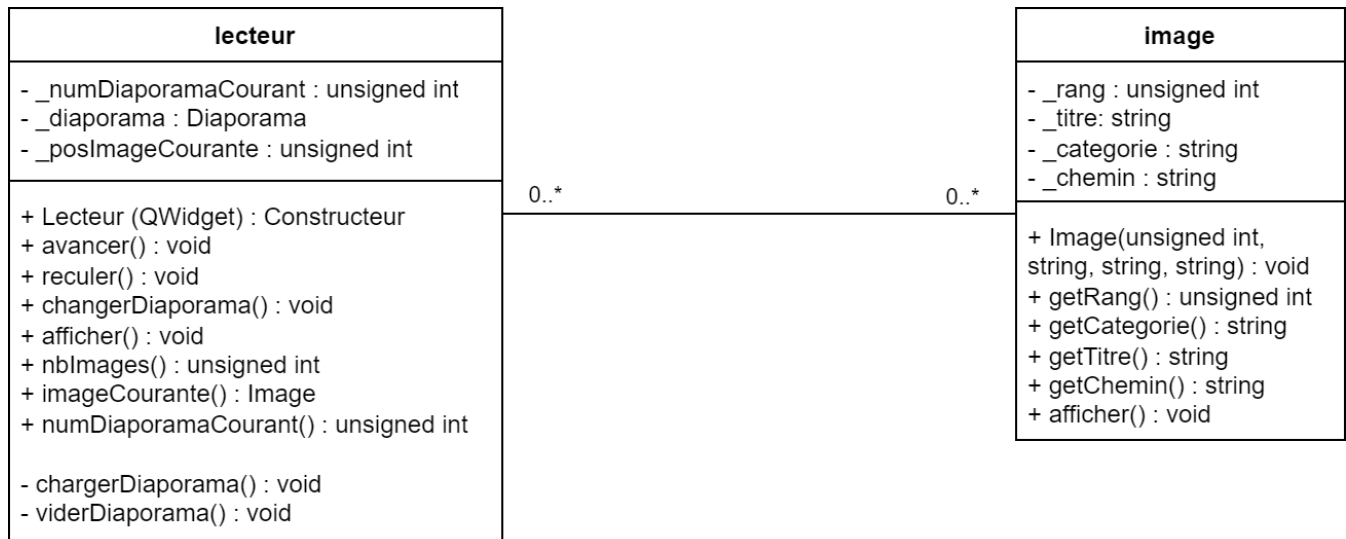


2. Scénarios

Enchaînement Nominal		
Messages	Acteur : Utilisateur	Système : DiapoForge
1	L'utilisateur demande de charger le diaporama	
2		Le système charge le diaporama
3		Le système affiche le diaporama
4	L'utilisateur demande à passer à la diapositive suivante	
5		Le système affiche la diapositive suivante
6	L'utilisateur demande à passer à la diapositive précédente	
7		Le système affiche la diapositive précédente
Enchaînements Alternatifs		
	Acteur : Utilisateur	Système : DiapoForge
4.A	Mode Auto	
4.A.1	L'utilisateur demande à déclencher le mode automatique	
4.A.2		Le système déclenche le mode automatique
	Acteur : Utilisateur	Système : DiapoForge
4.B	Changement vitesse	
4.B.1	L'utilisateur demande à modifier la vitesse d'affichage du diaporama	
4.B.2		Le système modifie la vitesse d'affichage du diaporama

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 LectureVue			
Nom attribut	Signification	Type	Exemple
ui	Interface graphique de la class LectureVue	ui	LectureVue.ui
_numDiaporamaCourant	Le numéro du diaporama en cours d’affichage	unsigned int	1
_diaporama	Pointeur vers une instance du vecteur Diaporama	Diaporama	
_posImageCourante	Position de l’image en cours d’affichage	unsigned int	4
Classe Image			
Nom attribut	Signification	Type	Exemple
_rang	Le rang de l'image dans le diapo	unsigned int	4
_titre	Le titre de l'image	string	"Bâtiment"
_categorie	La catégorie dans laquelle se trouve l'image	string	"Château Cendrillon"
_chemin	Le chemin d'accès de l'image	string	"../QtProjects/SAE201-Diaporama/cartesDisney/Disney_0.gif"

Tableau 2 : Dictionnaire des éléments v0

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

Exemple (classe lecteur de la version Console) :

Lecteur.h

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

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

class Lecteur
{
public:
    Lecteur();
    void avancer();                  // incrémente _posImageCourante, modulo nbImages()
    void reculer();                  // décrémente _posImageCourante, modulo nbImages()
    void changerDiaporama(unsigned int pNumDiaporama);    // permet de choisir un diaporama, 0 si
    // aucun diaporama souhaité
    void afficher();                  // affiche les informations sur lecteur-diaporama et image
    // courante
    unsigned int nbImages();          // affiche la taille de _diaporama
    Image* imageCourante();           // retourne le pointeur vers l'image courante
    unsigned int numDiaporamaCourant();

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

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

#endif // LECTEUR_H
```

Image.h

```
#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
```

Figure 4 : Schéma de classes = Classe XXX

Version v0 – Version console seule

4. Implémentation et tests

4.1 Implémentation

Liste et rôle des fichiers de cette version :

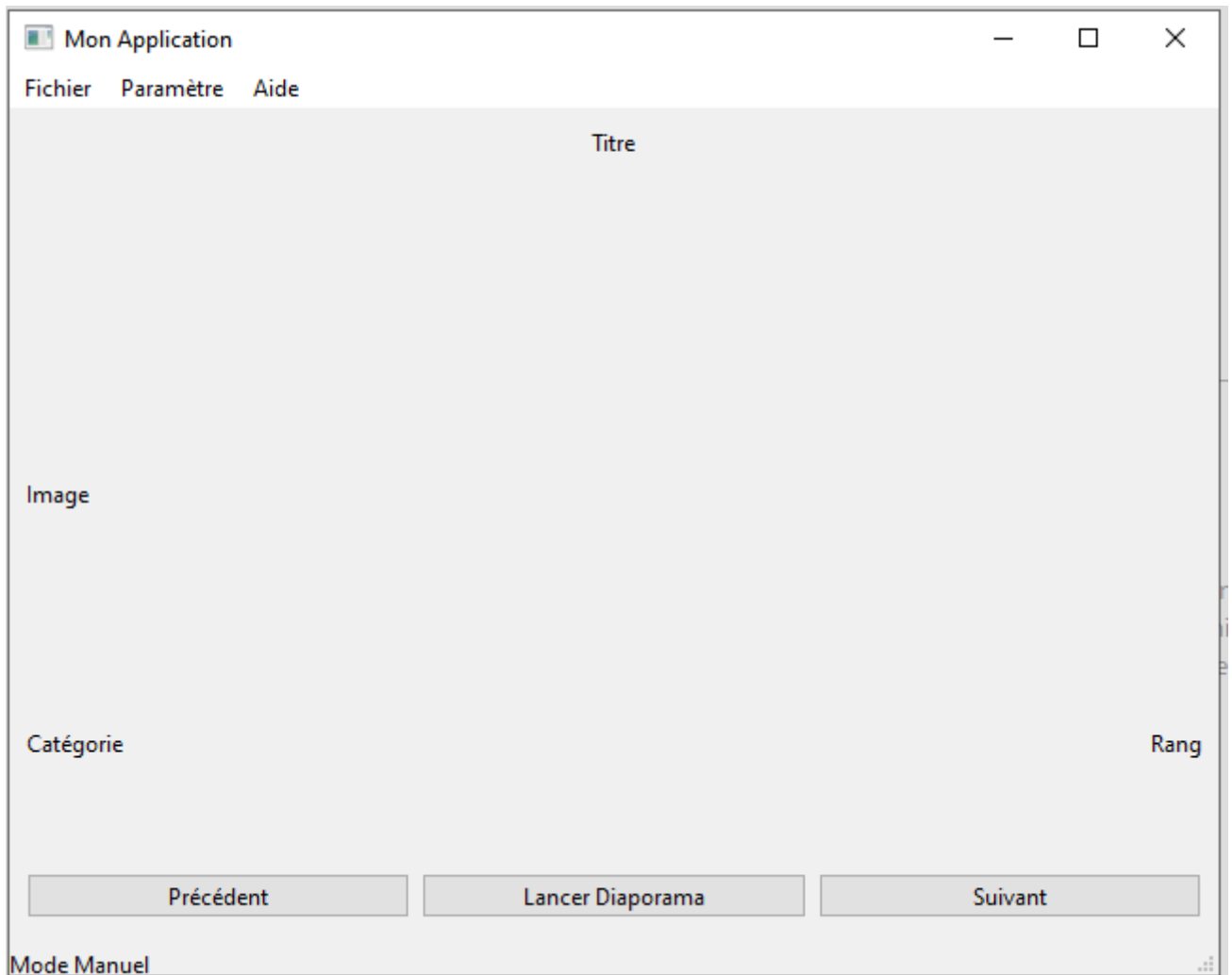
lecteurvue.h	Spécification de la classe Lecteur
lecteurvue.cpp	Corps de la classe Lecteur
image.h	Spécification de la classe Image
image.cpp	Corps de la classe Image
main.cpp	Tester les méthodes de la classe Lecteur

4.2 Test

Test avec le programme fournit main.cpp

Classe	Description	Valeur(s) en entrée	Résultat(s) attendu(s)	Résultat(s) obtenu(s)
valide n°1	Le chemin d'accès correspond à une image d'extension .gif	"C:\\cartesDisney\\carteDisney_0.gif"	C:\\cartesDisney\\carteDisney_0.gif	C:\\cartesDisney\\carteDisney_0.gif
valide n°2	La catégorie de l'image est contenu dans les valeurs possibles ("personne", "animal" ou "objet")	"animal"	animal	animal
valide n°3	Le chemin d'accès correspond à une image d'extension .png	"C:\\cartesDisney\\carteDisney_0.png"	C:\\cartesDisney\\carteDisney_0.png	C:\\cartesDisney\\carteDisney_0.png
invalide n°1	Le chemin d'accès ne correspond à aucune image	"C:\\cartesDisney\\carteDisney_70.gif"	Echec	Echec
invalide n°2	Le chemin d'accès correspond à un fichier texte	"C:\\cartesDisney\\carteDisney_0.txt"	Echec	Echec
invalide n°3	Le chemin d'accès est vide	""	Echec	Echec
invalide n°4	La catégorie de l'image n'est pas contenue dans les valeurs possibles	"nourriture"	Echec	Echec
invalide n°5	La catégorie de l'image est vide	""	Echec	Echec
invalide n°6	Le titre de l'image est vide	""	Echec	Echec

5. Éléments d'interface



Description sommaire des éléments de l'interface

- Titre : Le titre de l'image courante.
- Image : placeholder de l'image.
- Catégorie : Catégorie de l'image courante (personne, animal, etc...).
- Rang : Rang de l'image courante.
- Bouton "Précédent" : Permet de passer à l'image précédente.
- Bouton "Lancer Diaporama" : Permet de lancer le mode automatique.
- Bouton "Suivant" : Permet de passer à l'image suivante.
- Le menu " Fichier " : Permet d'accéder au menu "Sortir".
- Le menu " Fichier -> Sortir " : Fera quitter l'application.
- Le menu " Paramètres " : Permet d'accéder aux menus "Changer diaporama", "Enlever diaporama" et "Vitesse de défilement".
- Le menu " Paramètres -> Charger diaporama " permet de sélectionner le diaporama souhaité dans une base de données. L'application charge alors en mémoire le contenu du diaporama, c'est-à-dire son intitulé, ainsi que la liste des images qui lui sont associées.
- Le menu " Paramètres -> Enlever diaporama " permet de supprimer du lecteur le diaporama en cours de visualisation. Le lecteur revient alors à son état initial, vide.

- Le menu “ Paramètres -> Vitesse de défilement » permet de choisir le temps d’affichage de chaque image (en secondes) lorsque le diaporama est en mode Auto. Le temps d’affichage est propre à chaque Diaporama.
- Le menu “ Fichier -> Quitter “ permet de d’arrêter l’application, en ayant préalablement supprimé du lecteur le diaporama en cours de visualisation.
- Le menu “ Aide -> A propos de... “ ouvre une boîte de Message indiquant les auteurs de l’application et la version de l’application.

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 :

Nous avons agencé l’interface de la manière suivante :

- La barre de menu, dont la position est gérée par l’OS, qui contient des boutons nécessaires à certaines fonctionnalités métier.
- Le label indiquant le titre de l’image actuellement affichée, placé au-dessus de l’image.
- Le label image, qui accueillera le QPixmap de l’image
- les labels catégorie et rang, indiquant respectivement ces informations concernant l’image actuelle.
- Les boutons Précédent et Suivant qui permettent la navigation manuelle du diaporama
- Le bouton toggleAuto qui permet soit l’activation soit la désactivation du mode Automatique, et dont le texte sera adapté selon l’état de l’application
- Le label lEtat dans la statusBar qui indique l’état actuel de l’application (“Déchargé”, “Manuel” ou “Automatique”).

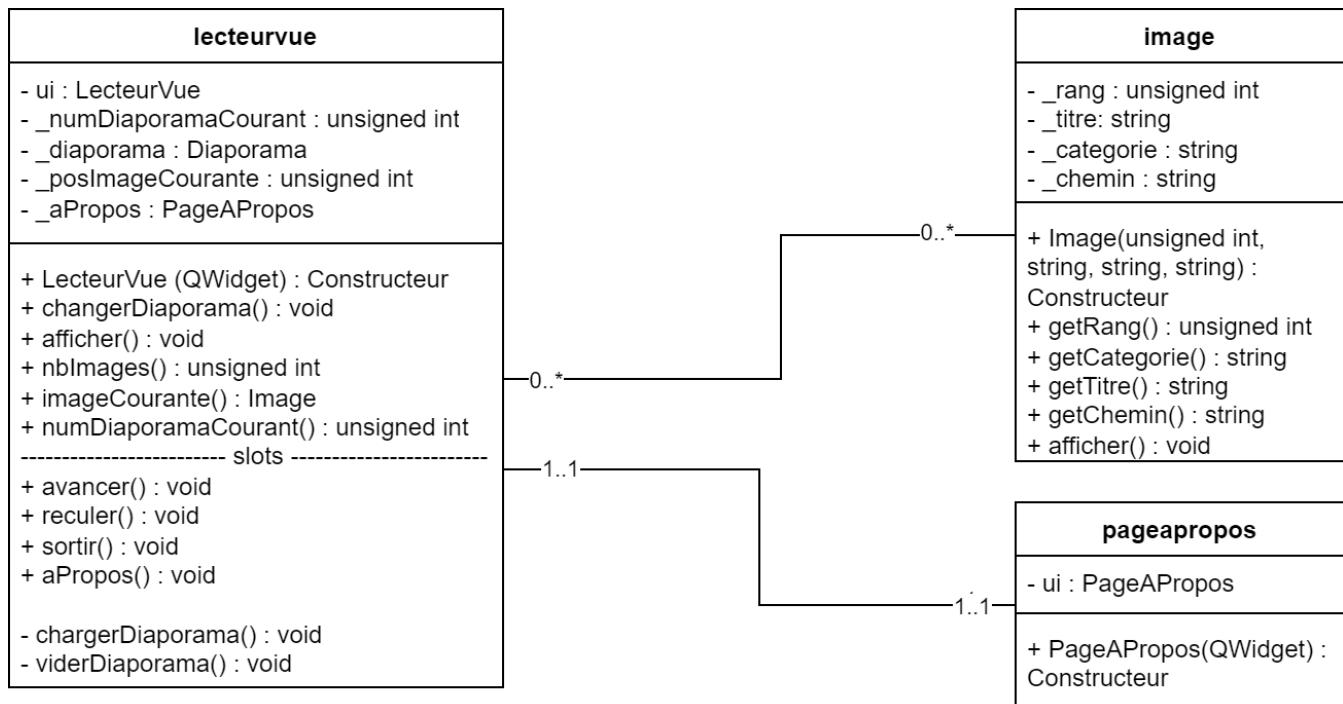
6.2 Test

Slots	Description	Résultat(s) attendu(s)	Résultat(s) obtenu(s)
avancer()	Affiche un message lorsque le bouton <i>Suivant</i> est cliqué	"Vous avez cliqué sur le bouton avancer."	"Vous avez cliqué sur le bouton avancer."
reculer()	Affiche un message lorsque le bouton <i>Précédent</i> est cliqué	"Vous avez cliqué sur le bouton reculer."	"Vous avez cliqué sur le bouton reculer."
demarrerAuto()	Affiche un message lorsque le bouton <i>Lancer Diaporama</i> est cliqué	"Vous avez cliqué sur le bouton démarrer le diapo."	"Vous avez cliqué sur le bouton démarrer le diapo."
arreterAuto()	Affiche un message lorsque le bouton <i>Arrêter Diaporama</i> est cliqué	"Vous avez cliqué sur le bouton arrêter le diapo."	"Vous avez cliqué sur le bouton arrêter le diapo."
sortir()	Affiche un message lorsque le	"Vous avez cliqué sur le	"Vous avez cliqué sur le

	bouton <i>Sortir</i> est cliqué	bouton sortir."	bouton sortir."
aPropos()	Affiche un message lorsque le bouton <i>A propos de</i> est cliqué	"Vous avez cliqué sur le bouton à propos."	"Vous avez cliqué sur le bouton à propos."
changerVitesse()	Affiche un message lorsque le bouton <i>Vitesse de défilement</i> est cliqué	"Vous avez cliqué sur le bouton changer vitesse."	"Vous avez cliqué sur le bouton changer vitesse."
demarrerAuto()	Affiche un message lorsque le bouton <i>Vitesse de défilement</i> est cliqué	"Vous avez cliqué sur le bouton démarrer le diapo."	"Vous avez cliqué sur le bouton démarrer le diapo."
chargerDiaporama()	Affiche un message lorsque le bouton <i>Charger le diaporama</i> est cliqué	"Vous avez cliqué sur le bouton charger diaporama."	"Vous avez cliqué sur le bouton charger diaporama."
viderDiaporama()	Affiche un message lorsque le bouton <i>Enlever le diaporama</i> est cliqué	"Vous avez cliqué sur le bouton vider le diapo."	"Vous avez cliqué sur le bouton vider le diapo."

7. Diagramme de classes (UML)

(a) Le diagramme de classes UML



(b) Dictionnaire des éléments pour chaque classe

Classe LectureVue			
Nom attribut	Signification	Type	Exemple
ui	Interface graphique de la class LectureVue	ui	LectureVue.ui
_numDiaporamaCourant	Le numéro du diaporama en cours d’affichage	unsigned int	1
_diaporama	Pointeur vers une instance du vecteur Diaporama	Diaporama	
_posImageCourante	Position de l’image en cours d’affichage	unsigned int	4
_aPropos	Pointeur vers une instance de la classe PageAPropos	PageAPropos	
Classe Image			
Nom attribut	Signification	Type	Exemple
_rang	Le rang de l'image dans le diapo	unsigned int	4
_titre	Le titre de l'image	string	"Bâtiment"
_categorie	La catégorie dans laquelle se trouve l'image	string	"Château Cendrillon"
_chemin	Le chemin d'accès de l'image	string	"../QtProjects/SAE201-Diaporama/cartesDisney/Disney_0.gif"
Classe PageAPropos			

Nom attribut	Signification	Type	Exemple
ui	Interface graphique de la class PageAPropos	ui	PageAPropos.ui

Tableau 2 : Dictionnaire des éléments v2

8. Comportement de l'application

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

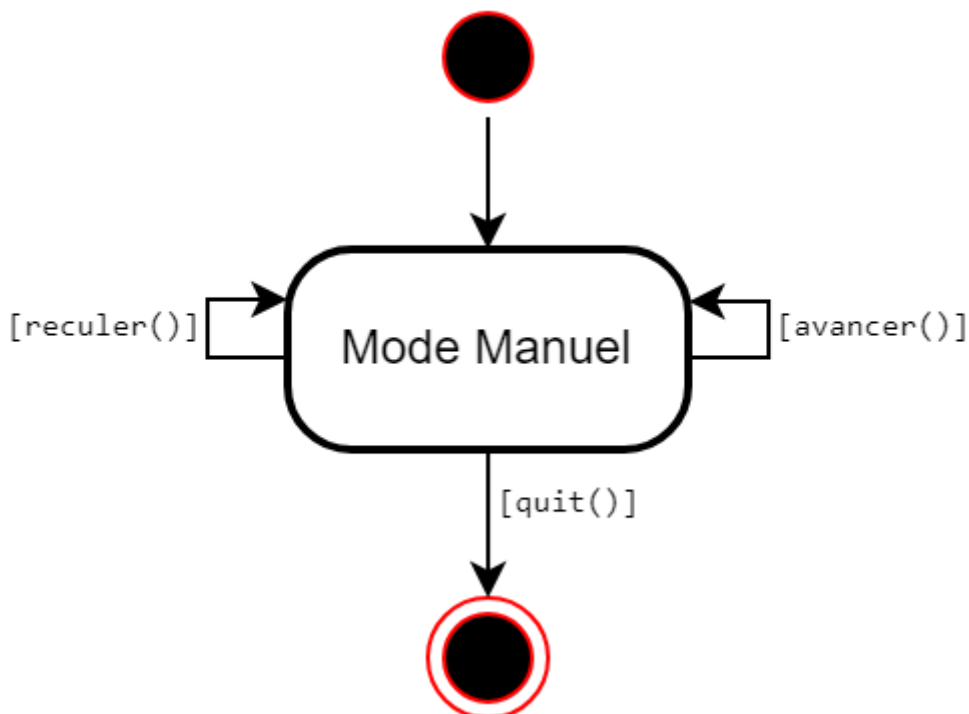


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v2

8.2 Dictionnaire des états, événements et Actions (v2)

Dictionnaire des états du diaporama

<i>nomEtat</i>	<i>Signification</i>
Mode Manuel	L'application affiche le diaporama en mode manuel, c.a.d. il faut passer manuellement à la prochaine ou la précédente diapositive.

Tableau 2 : États du lecteur de diaporamas – v2

Dictionnaire des événements faisant changer le diaporama d'état

<i>nomÉvénement</i>	<i>Signification</i>
[avancer()]	Le diaporama passe à l'image suivante
[reculer()]	Le diaporama passe à l'image précédente
[Quit()]	L'application se ferme

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

Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>
avancer sur le diaporama	L'utilisateur indique qu'il veut avancer sur le diaporama
reculer sur le diaporama	L'utilisateur indique qu'il veut reculer sur le diaporama
sortir	L'utilisateur clique sur le bouton Fichier-> Sortir L'utilisateur clique sur le bouton pour fermer la fenêtre

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

8.3 Table $T_{EtatsEvenementsActions}$ (v2)

Élément graphique prenant en charge cet événement <input type="checkbox"/>	bpSuivant	bpPrecedent	actionSortir
Événement <input type="checkbox"/> <i>nomEtat</i> ↓	Avancer sur le diaporama	Reculer sur le diaporama	sortir
Mode Manuel	Mode Manuel	Mode Manuel	Mode Manuel

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

9. Implémentation et tests

9.1 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
image.h	Spécification de la classe Image
image.cpp	Corps de la classe Image
pageapropos.h	Spécification de la classe graphique Qt contenant l'interface de la page "À propos"
pageapropos.cpp	Corps de la classe PageAPropos
pageapropos.ui	Fichier du dessin de l'interface réalisé par QtDesigner
main.cpp	??

Remarques sur l'implémentation :

Nous avons implémenté la page "à propos" dans une classe dédiée pour contrôler finement son apparence et pouvoir la changer avec aisance.

9.2 Tests (v2)

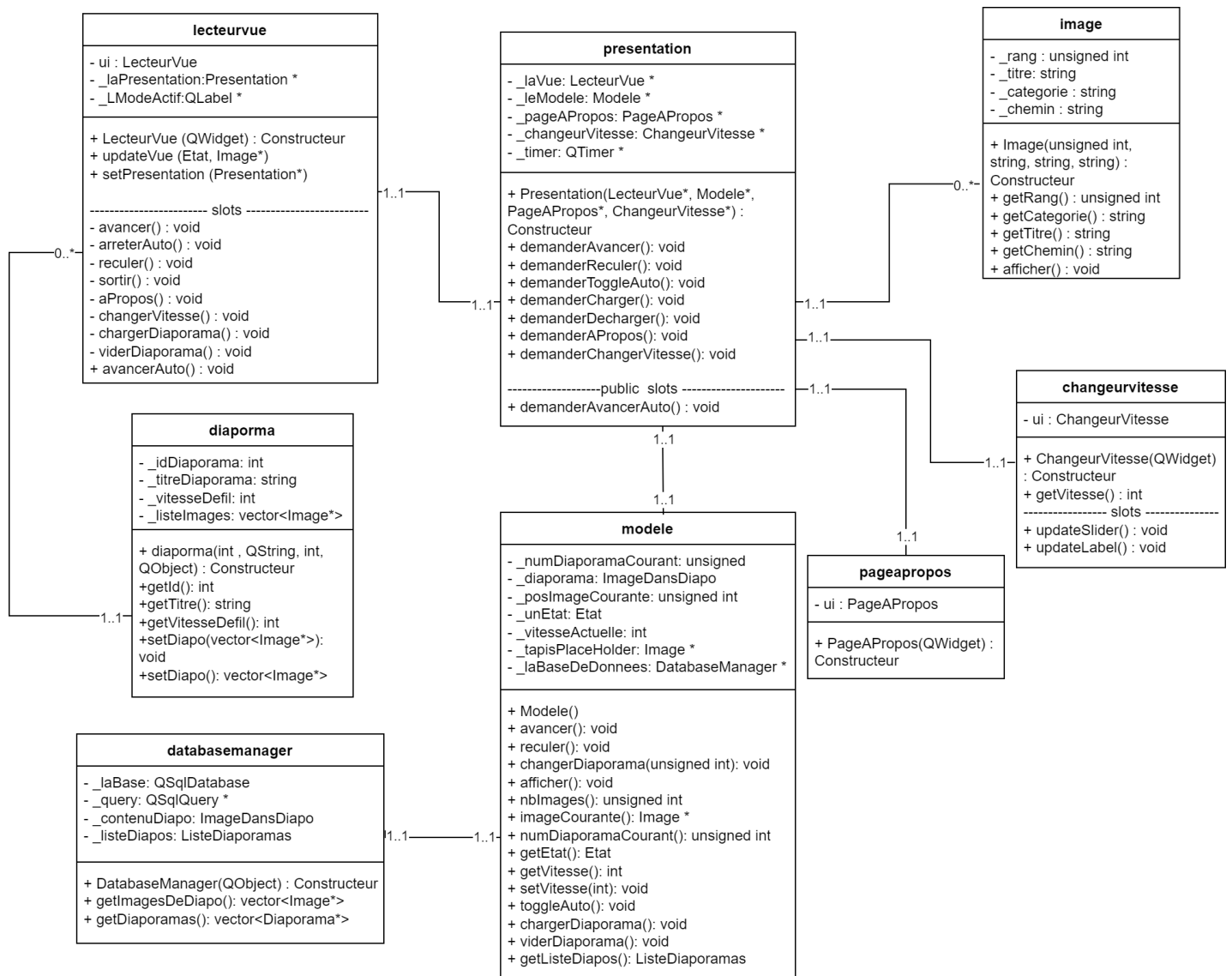
Classe	Description	Valeur(s) en entrée	Résultat(s) attendu(s)	Résultat(s) obtenu(s)
valide n°1	Le chemin d'accès correspond à une image d'extension .gif	"C:\\cartesDisney\\carteDisney_0.gif"	Affichage de l'image dans l'interface	Affichage de l'image dans l'interface

valide n°2	La catégorie de l'image est contenu dans les valeurs possibles ("personne", "animal" ou "objet")	"animal"	Affichage de la catégorie dans l'interface	Affichage de la catégorie dans l'interface
valide n°3	Le chemin d'accès correspond à une image d'extension .png	"C:\\cartesDisney\\carteDisney_0.png"	Affichage de l'image dans l'interface	Affichage de l'image dans l'interface
valide n°4	L'utilisateur clique sur le bouton <i>Suivant</i>	Le bouton <i>pbSuivant</i> est activé	Défilement à l'image suivante et affichage des informations dans la console	Défilement à l'image suivante et affichage des informations dans la console
valide n°5	L'utilisateur clique sur le bouton <i>Précédent</i>	Le bouton <i>pbPrecedent</i> est activé	Retour à l'image précédente et affichage des informations dans la console	Retour à l'image précédente et affichage des informations dans la console
valide n°6	L'utilisateur clique sur le bouton <i>Démarrer Diaporama</i>	Le bouton <i>pbToggleAuto</i> est activé	Changement du statut en mode automatique et affichage d'un message dans la console	Changement du statut en mode automatique et affichage d'un message dans la console
valide n°7	L'utilisateur clique sur le bouton <i>Arrêter Diaporama</i>	Le bouton <i>pbToggleAuto</i> est activé	Changement du statut du diaporama en mode manuel et affichage d'un message dans la console	Changement du statut du diaporama en mode manuel et affichage d'un message dans la console
valide n°8	L'utilisateur clique sur le bouton <i>Quitter</i>	L'action <i>actionSortir</i> est appelée	Fermeture de la fenêtre	Fermeture de la fenêtre
valide n°9	L'utilisateur clique sur le bouton <i>A propos de</i>	L'action <i>action_propos_de</i> est appelée	Affichage d'une fenêtre contenant la version de l'application, la date de création et les auteurs	Affichage d'une fenêtre contenant la version de l'application, la date de création et les auteurs
valide n°10	L'utilisateur clique sur le bouton <i>Vitesse de défilement</i>	L'action <i>actionChanger_diaporama</i> est appelée	Affichage d'un message dans la console	Affichage d'un message dans la console
valide n°11	L'utilisateur clique sur le bouton <i>Enlever le diaporama</i>	L'action <i>actionEnlever_diaporama</i> est appelée	Affichage d'un message dans la console et vide le diaporama	Affichage d'un message dans la console et vide le diaporama
valide n°12	L'utilisateur clique sur le bouton <i>Charger le diaporama</i>	L'action <i>actionChanger_diaporama</i> est appelée	Affichage d'un message dans la console et changement du diaporama	Affichage d'un message dans la console et changement du diaporama
invalide n°1	Le chemin d'accès ne correspond à aucune image	"C:\\cartesDisney\\carteDisney_70.gif"	Echec	Echec
invalide n°2	Le chemin d'accès correspond à un fichier texte	"C:\\cartesDisney\\carteDisney_0.txt"	Echec	Echec
invalide n°3	Le chemin d'accès est vide	""	Echec	Echec

invalide n°4	La catégorie de l'image n'est pas contenue dans les valeurs possibles	"nourriture"	Echec	Echec
invalide n°5	La catégorie de l'image est vide	""	Echec	Echec
invalide n°6	Le titre de l'image est vide	""	Echec	Echec

Version v5 –

10. Diagramme de classes (UML)



Ce diagramme concerne la version 5 du programme diaporama, il faut savoir que cette version suit déjà le modèle MVP

Classe LectureVue			
Nom attribut	Signification	Type	Exemple
ui	Interface graphique de la classe LectureVue	ui	LectureVue.ui
_laPresentation	Pointeur vers la classe intermédiaire entre la Vue et le Modèle. Élément Présentation du patron MVP.	Presentation	
_LModeActif	Élément graphique de la barre de status. Déclaré dans le code, à cause d'une incapacité à le placer dans la barre de status depuis Qt Design Studio.	QLabel	
Classe Image			
Nom attribut	Signification	Type	Exemple
_rang	Le rang de l'image dans le diapo	unsigned int	4
_titre	Le titre de l'image	string	"Bâtiment"
_categorie	La catégorie dans laquelle se trouve l'image	string	"Château Cendrillon"
_chemin	Le chemin d'accès de l'image	string	"../QtProjects/SAE201-Diaporama/cartesDisney/Disney_0.gif"
Classe ChangeurVitesse			
Nom attribut	Signification	Type	Exemple
ui	Interface graphique de la class ChangeurVitesse	ui	ChangeurVitesse.ui
Classe PageAPropos			
Nom attribut	Signification	Type	Exemple
ui	Interface graphique de la class PageAPropos	ui	PageAPropos.ui
Classe Diaporama			
Nom attribut	Signification	Type	Exemple
_idDiaporama	Identifiant du diaporama	int	PageAPropos.ui
_titreDiaporama	Titre du diaporama	string	1
_vitesseDefil	Vitesse de défilement de ce diaporma (en sec)	int	3
_listeImages	Listes des images contenues dans le diaporama	vector<Image*>	
Classe Presentation			
Nom attribut	Signification	Type	Exemple
_laVue	Pointeur vers l'interface de l'application. Élément Vue du modèle MVP.	LecteurVue	
_leModele	Pointeur vers la partie métier de l'application. Élément Modèle du modèle MVP.	Modele	
_pageAPropos	Pointeur vers la fenêtre "À propos" de l'application. Élément Vue du modèle MVP.	PageAPropos	
_changeurVitesse	Pointeur vers la fenêtre permettant le changement de la vitesse. Élément Vue du modèle MVP.	ChangeurVitesse	
_timer	Pointeur vers le QTimer qui gère le passage automatique des images en mode automatique.	QTimer	

Classe Modele			
Nom attribut	Signification	Type	Exemple
_numDiaporamaCourant	Identifiant du diaporama (idDiaporama dans la base de donnée) actuellement chargé.	unsigned int	2
_diaporama	Vecteur d'images du diaporama actuellement chargé.	ImageDansDiapo	vector<Image>
_posImageCourante	Position actuelle dans le diaporama	unsigned int	3
_unEtat	État actuel de l'application (Déchargé, Manuel ou Automatique)	Etat	Modele::Manuel
_vitesseActuelle	Vitesse de passage des images en mode auto.	int	3
_tapisPlaceholder	Image affichée lorsque le diaporama est déchargé.	Image	Image(0, "Catégorie", "Titre", PathTapis);
_laBaseDeDonnees	Gestionnaire de la base de données permettant les récupération des données	DatabaseManager	
Classe Databasemanager			
Nom attribut	Signification	Type	Exemple
_laBase	Élément Qt pour connecter la base de données.	QSqlDatabase	
_query	Élément Qt pour les requêtes.	QSqlQuery	
_contenuDiapo	Vecteur d'images du diaporama à charger.	ImageDansDiapo	
_listeDiapos	Vecteur de diaporamas à afficher pour le choix de l'utilisateur.	ListeDiaporamas	

11. Comportement de l'application

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

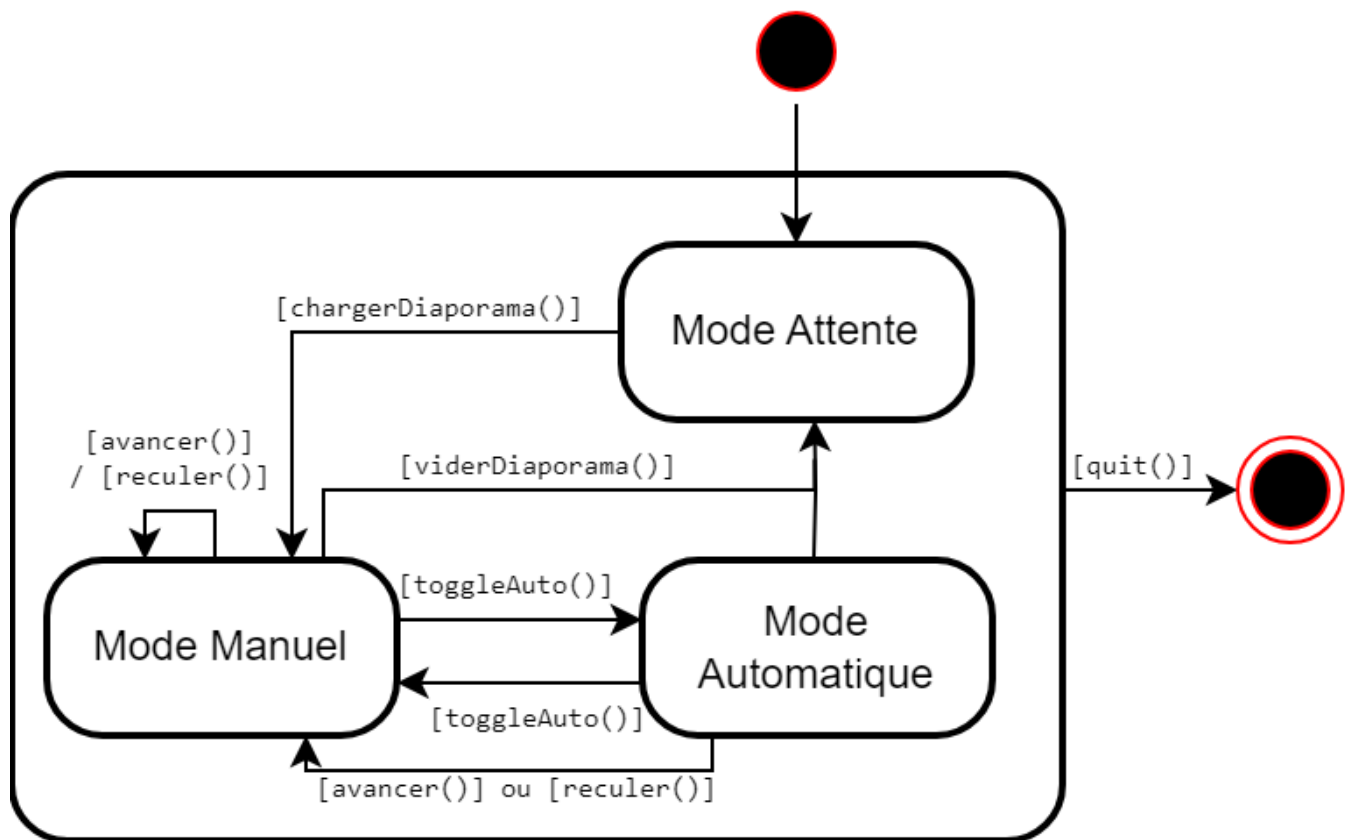


Figure 9 : Diagramme états-transitions du lecteur de diaporamas – v5

11.2 Dictionnaire des états, événements et Actions (v5)

Dictionnaire des états du diaporama

nomEtat	Signification
Mode Attente	L'application affiche une image par default (placeholder) qui signifie que le diaporama n'est pas chargé, elle attend donc que l'utilisateur demande à charger le diaporama.
Mode Manuel	L'application affiche le diaporama en mode manuel, c.a.d. il faut passer manuellement à la prochaine ou la précédente diapositive.
Mode Automatique	L'application affiche le diaporama image par image à une vitesse défini.

Tableau 2 : États du lecteur de diaporamas – v5

Dictionnaire des événements faisant changer le diaporama d'état

nomEvénement	Signification
[chargerDiaporama()]	L'application charge le diaporama
[viderDiaporama()]	L'application décharge le diaporama
[toggleAuto()] Mode Manuel -> Mode Auto	L'application passe en mode auto

[toggleAuto()] Mode auto -> Mode Manuel	L'application passe en mode manuel
[avancer()] ou [reculer()]	L'application passe en mode manuel lorsque l'utilisateur indique qu'il veut avancer ou reculer dans la diaporama
[avancer()]	Le diaporama passe à l'image suivante
[reculer()]	Le diaporama passe à l'image précédente
[Quit()]	L'application se ferme

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

Description des actions réalisées lors de la traversée des transitions

<i>nomAction</i>	<i>Signification</i>
Charger le diaporama	L'utilisateur indique qu'il veut charger du diaporama
Vider le diaporama	L'utilisateur indique qu'il veut décharger du diaporama
Passage Auto	L'utilisateur clique sur le bouton Lancer Diaporama
Passage Manuel	L'utilisateur clique sur le bouton Arrêter Diaporama
Passage Manuel par bouton avancer ou reculer	L'utilisateur clique sur le bouton Suivant ou Précédent
avancer sur le diaporama	L'utilisateur indique qu'il veut avancer sur le diaporama
reculer sur le diaporama	L'utilisateur indique qu'il veut reculer sur le diaporama
sortir	L'utilisateur clique sur le bouton Fichier-> Sortir L'utilisateur clique sur le bouton pour fermer la fenêtre

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

11.3 Table T_EtatsEvenementsActions (v5)

Élément graphique prenant en charge cet événement □	actionChanger_diaporama	actionEnlever_diaporama	pbToggleAuto	pbToggleAuto	pbPrécédent ou pbSuivant	bpSuivant	bpPrécédent	actionSortir
Événement □ nomEtat↓	Charger le diaporama	Vider le diaporama	Passage Auto	Passage Manuel	Passage Manuel par bouton avancer ou reculer	Avancer sur le diaporama	Reculer sur le diaporama	sortir
Mode Attente	Mode Attente							Mode Attente
Mode Manuel		Mode Manuel	Mode Manuel			Mode Manuel	Mode Manuel	Mode Manuel
Mode Automatique		Mode Automatique		Mode Automatique	Mode Automatique			Mode Automatique

12. Implémentation et tests

12.1 Implémentation (v5)

Liste et rôle des fichiers de cette version :

lecteurvue.h	Spécification de la classe graphique Qt contenant l'interface du lecteur de diaporamas Partie Vue du patron MVP.
lecteurvue.cpp	Corps de la classe LecteurVue
lecteurvue.ui	Fichier du dessin de l'interface réalisé par QtDesigner
modele.h	Classe contenant les méthodes métiers de l'application. Partie Modèle du patron MVP.
modele.cpp	Corps de la classe Modele
presentation.h	Classe liant le modèle et la vue, gérant également les différentes fenêtres. Partie Présentation du patron MVP
presentation.cpp	Corps de la classe Presentation
databasemanager.h	Classe gérant la base de données et exécutant les requêtes avant de renvoyer les résultats au modèle.
databasemanager.cpp	Corps de la classe DatabaseManager
changeurvitesse.h	Spécification de la classe graphique Qt contenant l'interface du changeur de vitesse. Partie Vue du patron MVP.
changeurvitesse.cpp	Corps de la classe ChangeurVitesse
pageapropos.h	Spécification de la classe graphique Qt contenant l'interface de la page "À propos". Partie Vue du patron MVP.
pageapropos.cpp	Corps de la classe PageAPropos
image.h	Spécification de la classe Image Permet le stockage des informations sur les images.
image.cpp	Corps de la classe Image
main.cpp	Fichier déclarant les différents éléments et les liant entre eux.

Remarques sur l'implémentation :

L'utilisation des bases de données a été implémentée sous une classe à part, ainsi que les requêtes associées. La classe DatabaseManager exécute les requêtes et place les résultats dans des vecteurs avant de les donner au modèle pour être traités. Cela permet un remplacement simple des sources de données, comme par exemple l'utilisation de fichiers ne nécessiterait que des modifications mineures du Modèle.

12.2 Tests (v5)

Classe	Description	Valeur(s) en entrée	Résultat(s) attendu(s)	Résultat(s) obtenu(s)
--------	-------------	---------------------	------------------------	-----------------------

valide n°1	Le chemin d'accès correspond à une image d'extension .gif	"C:\\cartesDisney\\carteDisney_0.gif"	Affichage de l'image dans l'interface	Affichage de l'image dans l'interface
valide n°2	La catégorie de l'image est contenu dans les valeurs possibles ("personne", "animal" ou "objet")	"animal"	Affichage de la catégorie dans l'interface	Affichage de la catégorie dans l'interface
valide n°3	Le chemin d'accès correspond à une image d'extension .png	"C:\\cartesDisney\\carteDisney_0.png"	Affichage de l'image dans l'interface	Affichage de l'image dans l'interface
valide n°4	L'utilisateur clique sur le bouton <i>Suivant</i>	Le bouton <i>pbSuivant</i> est activé	Défilement à l'image suivante et affichage des informations dans la console	Défilement à l'image suivante et affichage des informations dans la console
valide n°5	L'utilisateur clique sur le bouton <i>Précédent</i>	Le bouton <i>pbPrecedent</i> est activé	Retour à l'image précédente et affichage des informations dans la console	Retour à l'image précédente et affichage des informations dans la console
valide n°6	L'utilisateur clique sur le bouton <i>Démarrer Diaporama</i>	Le bouton <i>pbToggleAuto</i> est activé	Changement du statut en mode automatique et affichage d'un message dans la console	Changement du statut en mode automatique et affichage d'un message dans la console
valide n°7	L'utilisateur clique sur le bouton <i>Arrêter Diaporama</i>	Le bouton <i>pbToggleAuto</i> est activé	Changement du statut du diaporama en mode manuel et affichage d'un message dans la console	Changement du statut du diaporama en mode manuel et affichage d'un message dans la console
valide n°8	L'utilisateur clique sur le bouton <i>Quitter</i>	L'action <i>actionSortir</i> est appelée	Fermeture de la fenêtre	Fermeture de la fenêtre
valide n°9	L'utilisateur clique sur le bouton <i>A propos de</i>	L'action <i>action_propos_de</i> est appelée	Affichage d'une fenêtre contenant la version de l'application, la date de création et les auteurs	Affichage d'une fenêtre contenant la version de l'application, la date de création et les auteurs
valide n°10	L'utilisateur clique sur le bouton <i>Vitesse de défilement</i>	L'action <i>actionChanger_diaporama</i> est appelée	Affichage d'un message dans la console	Affichage d'un message dans la console
valide n°11	L'utilisateur clique sur le bouton <i>Enlever le diaporama</i>	L'action <i>actionEnlever_diaporama</i> est appelée	Affichage d'un message dans la console et vide le diaporama	Affichage d'un message dans la console et vide le diaporama
valide n°12	L'utilisateur clique sur le bouton <i>Charger le diaporama</i>	L'action <i>actionChanger_diaporama</i> est appelée	Affichage d'un message dans la console et changement du diaporama	Affichage d'un message dans la console et changement du diaporama
invalid n°1	Le chemin d'accès ne correspond à aucune image	"C:\\cartesDisney\\carteDisney_70.gif"	Echec	Echec

invalide n°2	Le chemin d'accès correspond à un fichier texte	"C:\\cartesDisney\\carteDisney_0.txt"	Echec	Echec
invalide n°3	Le chemin d'accès est vide	""	Echec	Echec
invalide n°4	La catégorie de l'image n'est pas contenue dans les valeurs possibles	"nourriture"	Echec	Echec
invalide n°5	La catégorie de l'image est vide	""	Echec	Echec
invalide n°6	Le titre de l'image est vide	""	Echec	Echec

13. Bilan

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

<https://github.com/Feror-BotMaker/SAE201-Diaporama>

Temps global de travail (pour le groupe):

Nous avons travaillé environ 7h par personne (14h pour le groupe), nous ne sommes que 2 pour cette SAE.

Apprentissages majeurs:

Meilleure maîtrise de github et du versionnage

Capacité à corréler base de données et logiciel (même si cet apprentissage est fait avec odbc et donc pas forcément maîtrisé sur d'autre système d'exploitation du moins pour Maximilien, Jokin l'ayant dû l'importer sur sa machine personnelle sous Mac).

Difficultés majeures:

Difficultés à corriger les bases de données et qt, parce que Jokin a voulu l'utiliser sur sa machine personnelle.

Difficultés à réaliser des jeux de tests utiles et intéressants.

Points positifs / négatifs de l'activité:

-Les explications sur comment le dossier était à réaliser est à revoir, nous ne savons pas bien ce qu'il faut faire et obligé de demander de l'aide au professeur car certaines parties n'ont pas été vues (ie. les jeux de tests sur applications graphiques).

-Sujets originaux

Tableau Bilan

V0	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	A comprendre un code donné	Essayer de le comprendre le code	Le tri des images	Le suivant modulo	1h	30min	-	Détailler un peu plus le fonctionnement de la V0 dès le début de la SAÉ
Valentin DUMAS	À utiliser des classes créé sur QT	-	Comprendre et compléter un code que je n'ai pas créé	-	1h	-	-	Détailler un peu plus le fonctionnement de la V0 dès le début de la SAÉ
Jokin SUARES	-	A optimiser un code	-	Le tri des images	1h	1h30	-	Détailler un peu plus le fonctionnement de la V0 dès le début de la SAÉ
V1	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	Designer une interface à partir d'un texte faisant office de cahier des charges	Imaginer le design de l'application	-	La penser de manière logique / ergonomique	1h	-	-	-
Valentin DUMAS	Utiliser l'interface graphique de QT et créer une barre de menu	Designer l'interface de l'application	-	-	-	-	-	-
Jokin SUARES	-	-	-	-	-	30min	-	-
V2	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	Utiliser comme il se doit de la documentation pour mieux optimiser son code	Faire le lien entre la V0 et la V1	Le diagramme Etat transition	-	1h30	20min	-	-
Valentin	Adapter des méthodes d'une	-	-	-	-	30min	-	-

DUMAS	classe							
Jokin SUARES	Comment afficher une image dans Qt	Afficher une image dans Qt	-	-	15min	3h	-	
V3	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	Comment gérer un objet de classe Timer	Concevoir le Timer	-	La gestion de l'objet en lui-même	1h	30min	-	-
Valentin DUMAS	Créer et utiliser un timer	Utiliser un objet Timer	-	Arrêter le mode automatique	1h	-	-	-
Jokin SUARES	utilisation du QTimer	comprendre l'utilisation du QTimer	-	-	45min	1h	-	
V4	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	Comment mieux structurer son code pour les autres	Plus ou moins tout	-	Bien s'approprier le fonctionnement d'une boîte de dialogue	30min	2h	-	-
Jokin SUARES	Utilisation d'un regex pour limiter la saisie dans le changeur vitesse	lier le slider au champ de texte	-	-	15min	20min	-	
V4_MVP	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	Structurer l'application en patron MVP	Comprendre le Modèle MVP	-	Bien s'approprier le fonctionnement d'une boîte de	2h	20min	-	-

				dialogue				
Jokin SUARES	Structurer l'application en patron MVP	Comprendre le Modèle MVP	-	appliquer un patron MVP	20min	2h		Suggérer un modèle MVP dès le début
V5_MVP	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	interroger une base de données	réfléchir sur la construction des requêtes et leur application en c++	Diagramme de classe	Essayer de comprendre un code qui n'est pas le sien	30min	1h	-	-
Jokin SUARES	Mettre un serveur MySql en local sur MacOS	-	-	La correction de divers soucis liés aux requêtes	10min	45min	-	-
V6_MVP	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	Rajouter une classe "Diaporama"	-	-	-	45min	3h (pour l'instant)	-	-
Jokin SUARES	Lier un click sur une QTableWidgetItem à un champ de texte	Lier un click sur une QTableWidgetItem à un champ de texte	-	-	-	-	-	-
V7_MVP_B D	ce qu'on a appris	ce qu'on a aimé faire	ce qu'on a pas aimé faire	ce qui a été difficile	temps passé conception	temps passé code	ce qui aurait pu être mieux	ce qui pourrait être amélioré dans la SAÉ
Maximilien BODIN	-	-	-	-	-	10min	-	-
Jokin SUARES	-	Mettre des raccourcis clavier	-	-	-	30min	Que l'application ait un logo	