

ETCHECOPAR Maxime (TD2 – TP4)

LAGÜE Théo (TD2 – TP4)

PIGNEAUX Loris (TD2 – TP3)

## SAE S2.01

# Développement d'une application



## Table des matières

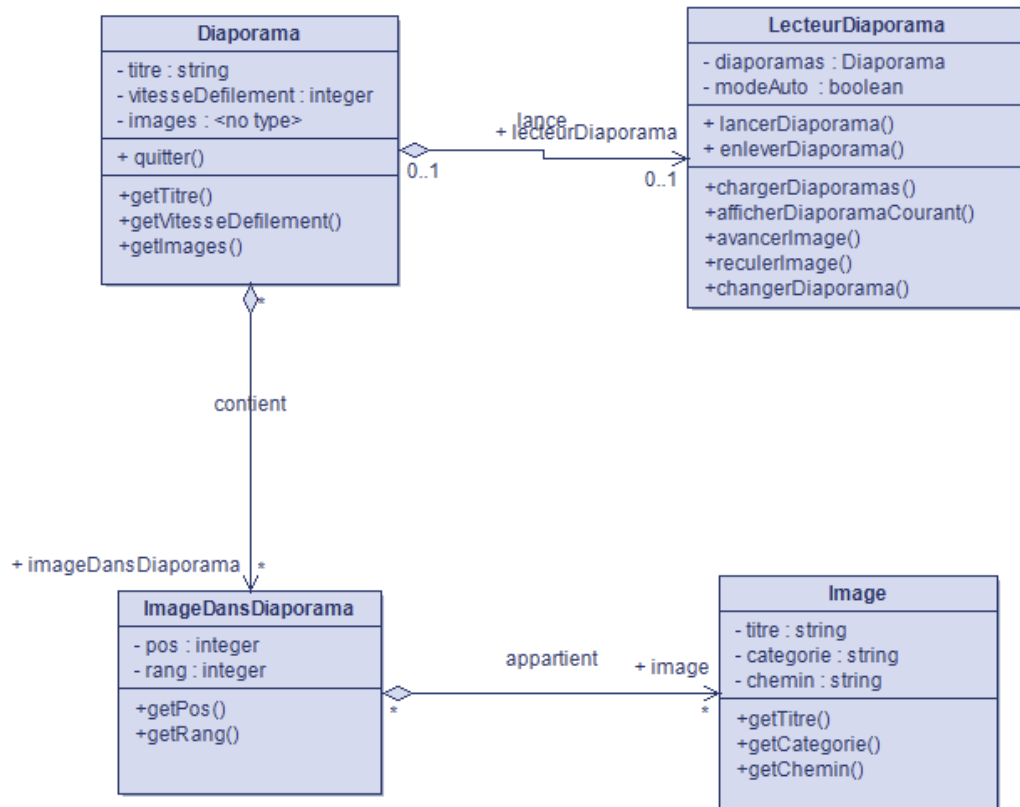
Rappel de la version : .....	3
Diagramme de classe : .....	4
1. Classe Diaporama .....	4
2. Classe LecteurDiaporama .....	4
3. Classe ImageDansDiaporama .....	5
4. Classe Image .....	5
Relations entre les classes : .....	5
• <b>Diaporama et LecteurDiaporama</b> : .....	5
• <b>Diaporama et ImageDansDiaporama</b> : .....	5
• <b>ImageDansDiaporama et Image</b> : .....	5
Rappel de la version : .....	7
Diagramme état-transition : .....	7
États .....	8
Transitions .....	9
Synthèse .....	9
Rappel de la version : .....	9
Diagramme état-transition : .....	10
États .....	10
Transitions .....	11
Synthèse .....	12
Rappel de la version : .....	12
Diagramme état-transition : .....	13
Bilan: .....	2

V1 :

## Rappel de la version :

v1	<p>Projet NON graphique Orienté Objets (= projet <b>console</b> non Qt).</p> <p>Version OO du programme précédent et conforme au diagramme de classes proposé. Le lecteur s'exécute en mode console et ne propose que le mode Manuel.</p> <p><b>A faire pour parvenir à cette version :</b></p> <p>Dossier de conception :</p> <ul style="list-style-type: none"><li>- Réaliser le diagramme des classes UML sous-jacent à l'application : classes Lecteur, ...</li><li>- Documenter les attributs et méthodes de chaque classe (signification / buts).</li></ul> <p>Code :</p> <ul style="list-style-type: none"><li>- Les classes figurant sur le diagramme UML sont implémentées.</li><li>- Le fichier main.cpp permet de tester le travail réalisé : il crée les principaux objets, affiche leur contenu, saisit les demandes de l'utilisateur, et active les méthodes des objets en fonction des demandes de l'utilisateur.</li></ul>
----	--

## Diagramme de classe :



### 1. Classe Diaporama

- **Attributs :**

- `titre : string` : Le titre du diaporama.
- `vitesseDefilement : integer` : La vitesse de défilement des images dans le diaporama.
- `images : <no type>` : Une liste d'images dans le diaporama (le type n'est pas spécifié ici).

- **Méthodes :**

- `+ quitter()` : Méthode pour quitter le diaporama.
- `+ getTitre()` : Retourne le titre du diaporama.
- `+ getVitesseDefilement()` : Retourne la vitesse de défilement des images.
- `+ getImages()` : Retourne la liste des images.

### 2. Classe LecteurDiaporama

- **Attributs :**

- `diaporamas : Diaporama` : Un objet de type Diaporama.
- `modeAuto : boolean` : Indique si le diaporama est en mode

automatique ou non.

- **Méthodes :**

- + `lancerDiaporama()` : Démarre le diaporama.
- + `enleverDiaporama()` : Arrête le diaporama.
- + `chargerDiaporamas()` : Charge les diaporamas disponibles.
- + `afficherDiaporamaCourant()` : Affiche le diaporama actuel.
- + `avancerImage()` : Passe à l'image suivante dans le diaporama.
- + `reculerImage()` : Revient à l'image précédente dans le diaporama.
- + `changerDiaporama()` : Change le diaporama affiché.

### 3. Classe ImageDansDiaporama

- **Attributs :**

- `pos : integer` : La position de l'image dans le diaporama.
- `rang : integer` : Le rang de l'image dans le diaporama.

- **Méthodes :**

- + `getPos()` : Retourne la position de l'image.
- + `getRang()` : Retourne le rang de l'image.

### 4. Classe Image

- **Attributs :**

- `titre : string` : Le titre de l'image.
- `categorie : string` : La catégorie de l'image.
- `chemin : string` : Le chemin d'accès à l'image.

- **Méthodes :**

- + `getTitre()` : Retourne le titre de l'image.
- + `getCategorie()` : Retourne la catégorie de l'image.
- + `getChemin()` : Retourne le chemin d'accès à l'image.

## Relations entre les classes :

- **Diaporama et LecteurDiaporama :**

- Un diaporama est lancé par un LecteurDiaporama. La relation est de type composition avec une cardinalité de 0..1 de chaque côté, indiquant qu'un diaporama peut être lancé par un LecteurDiaporama, mais ce n'est pas obligatoire.

- **Diaporama et ImageDansDiaporama :**

- Un diaporama contient plusieurs objets de type ImageDansDiaporama. La relation est de type agrégation, indiquant qu'un diaporama peut contenir plusieurs images (cardinalité 1..\*).

- **ImageDansDiaporama et Image :**

- Chaque ImageDansDiaporama est associé à une image. La relation est de type agrégation, indiquant qu'une image peut appartenir à plusieurs diaporamas (cardinalité 1..\*).

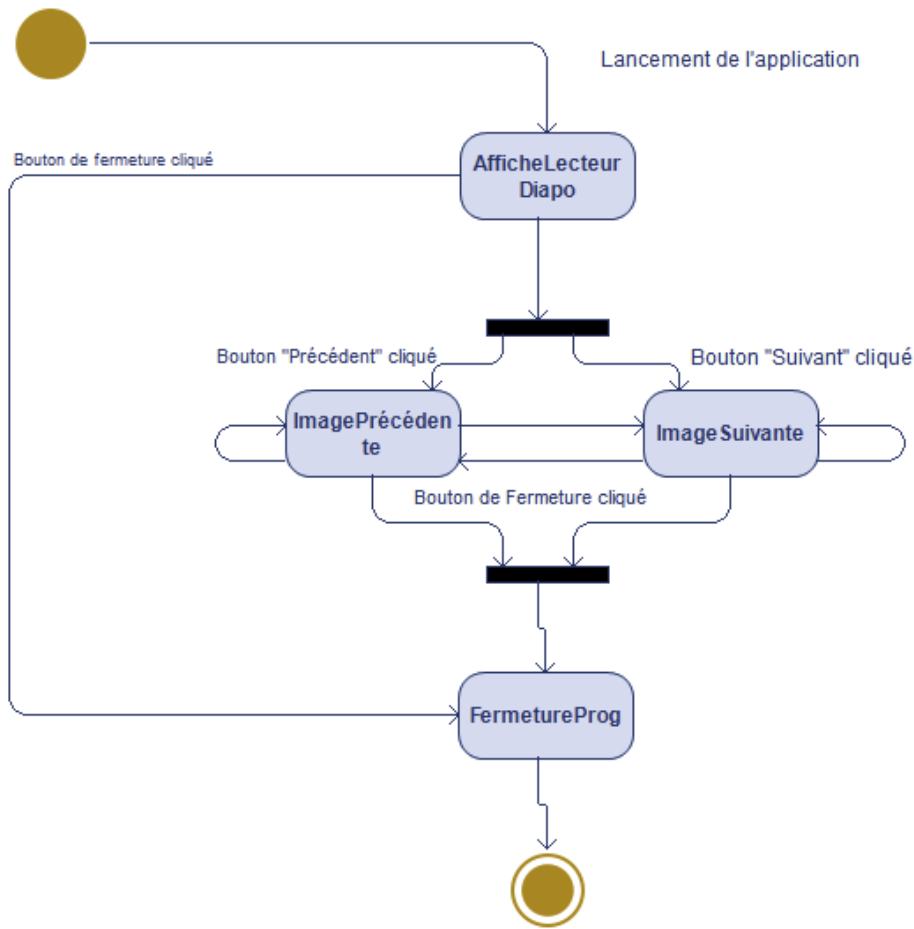


# V2 :

## Rappel de la version :

v2	<p>Projet Qt graphique.</p> <p>Seule l'interface graphique de l'application est complète et opérationnelle : tous les boutons et items de menus sont présents mais ils ne déclenchent pas les opérations métiers demandées : ils se contentent de communiquer (usage de <code>QDebug()</code>) qu'une action a été déclenchée par l'utilisateur.</p> <p><b>A faire pour parvenir à cette version :</b></p> <p>Dossier de conception :</p> <ul style="list-style-type: none"><li>- Réaliser le diagramme états-transitions de l'application uniquement (comportement du lecteur), sous ses 2 formes (classique matricielle)</li><li>- Documenter le lien entre éléments d'interface et fonctionnalités</li></ul> <p>Code :</p> <ul style="list-style-type: none"><li>- Création avec le générateur d'interfaces de la classe fenêtre personnalisée <code>LecteurVue</code> dans un projet Qt graphique : fichiers <code>lecteurvue.ui</code>, <code>lecteurvue.h</code>, <code>lecteurvue.cpp</code> et <code>main.cpp</code></li></ul>
----	--

## Diagramme état-transition :



## États

1. **Initial**
  - Représenté par le cercle plein (point de départ).
  - Lancement de l'application.
2. **AfficheLecteurDiapo**
  - Premier état après le lancement de l'application.
  - L'application affiche le lecteur de diaporama.
3. **ImagePrécédente**
  - État de l'application lorsque l'utilisateur clique sur le bouton "Précédent".
  - Affiche l'image précédente dans le diaporama.
4. **ImageSuivante**
  - État de l'application lorsque l'utilisateur clique sur le bouton "Suivant".
  - Affiche l'image suivante dans le diaporama.
5. **FermetureProg**
  - État de l'application lorsque l'utilisateur clique sur le bouton de fermeture.
  - Prépare l'application pour se fermer.
6. **Final**
  - Représenté par le cercle avec un point au centre (fin du processus).
  - Indique que l'application s'est fermée.



## Transitions

1. **Lancement de l'application**
  - Transition de l'état initial vers l'état AfficheLecteurDiapo.
2. **Bouton "Précédent" cliqué**
  - Transition de l'état AfficheLecteurDiapo vers l'état ImagePrécédente.
3. **Bouton "Suivant" cliqué**
  - Transition de l'état AfficheLecteurDiapo vers l'état ImageSuivante.
4. **Bouton de fermeture cliqué**
  - Transition de l'état AfficheLecteurDiapo vers l'état FermetureProg.
  - Transition de l'état ImagePrécédente vers l'état FermetureProg.
  - Transition de l'état ImageSuivante vers l'état FermetureProg.
5. **Fermeture de l'application**
  - Transition de l'état FermetureProg vers l'état final.

## Synthèse

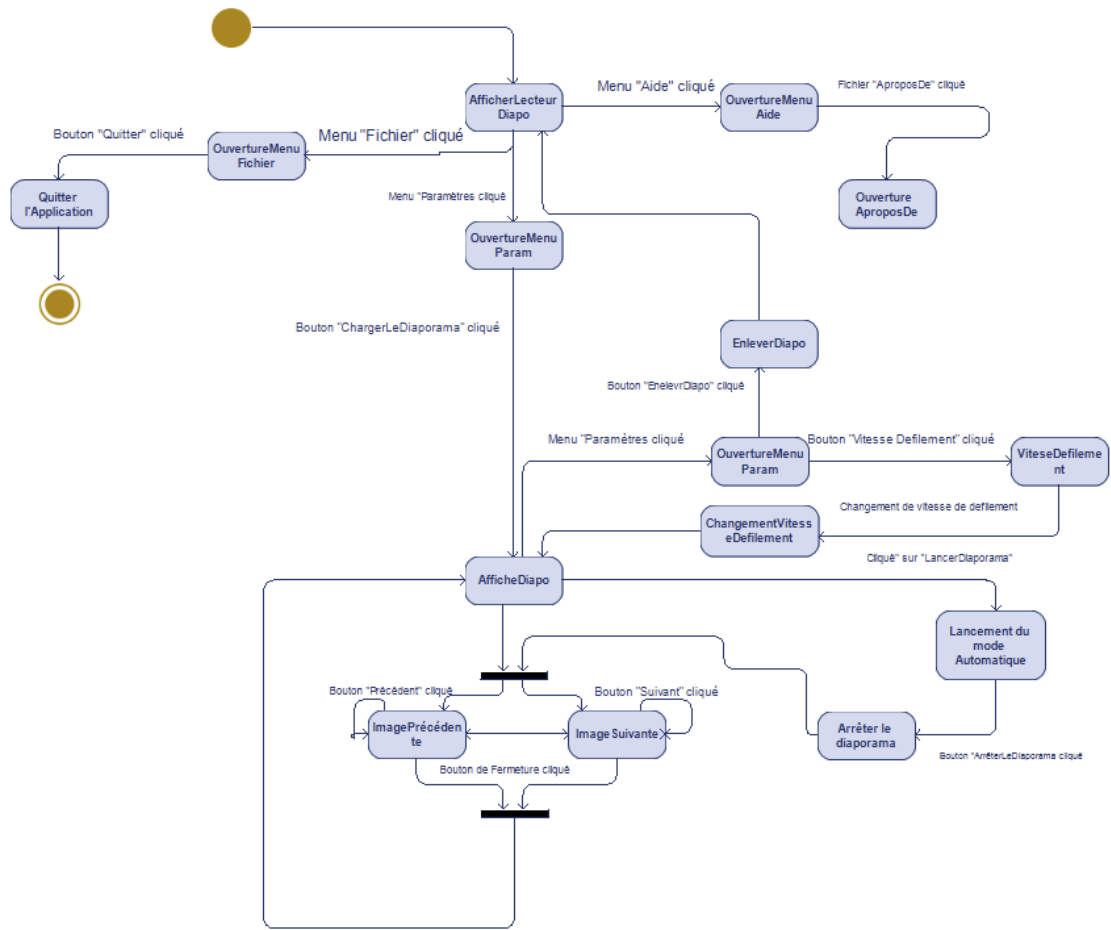
- Le diagramme commence par le lancement de l'application et passe à l'état AfficheLecteurDiapo.
- À partir de cet état, l'utilisateur peut naviguer entre les images en cliquant sur les boutons "Précédent" ou "Suivant".
- À tout moment, l'utilisateur peut cliquer sur le bouton de fermeture pour initier la fermeture de l'application, menant à l'état FermetureProg.
- L'état final est atteint lorsque l'application est fermée.

## V5 :

### Rappel de la version :

v5 / v5_MVP	<p>Projet créé par duplication de la v4 (ou v4_MVP)</p> <ul style="list-style-type: none"><li>- Implémentation de <a href="#">Fichier &gt;&gt; Paramétrer &gt;&gt; Vitesse ...</a> : ouvre une Boîte de Dialogue permettant de modifier la vitesse de défilement des images</li><li>- Au démarrage, aucun Diaporama n'est chargé</li><li>- Implémentation de <a href="#">Fichier &gt;&gt; Paramétrer &gt;&gt; Charger diaporama ...</a> : qui charge les images 'en dur' et les mets à disposition triées par ordre croissant de rang</li><li>- Implémentation de <a href="#">Fichier &gt;&gt; Paramétrer &gt;&gt; Enlever diaporama ...</a> : qui enlève le Diaporama en cours. Le lecteur est vide à nouveau.</li></ul> <p><b>A faire pour parvenir à cette version :</b></p> <p>Dossier de conception :</p> <ul style="list-style-type: none"><li>- Mise à jour de la description de la version</li></ul> <p>Code :</p> <ul style="list-style-type: none"><li>- Mise en œuvre de la fonctionnalité annoncée</li></ul>
-------------------	--

## Diagramme état-transition :



## États

1. **Initial**
  - Représenté par le cercle plein (point de départ).
2. **OuvertureMenuFichier**
  - État lorsque le menu "Fichier" est cliqué.
3. **AfficheLecteurDiapo**
  - État lorsque l'application affiche le lecteur de diaporama.
4. **OuvertureMenuAide**
  - État lorsque le menu "Aide" est cliqué.
5. **OuvertureAproposDe**
  - État lorsque l'option "À propos de" du menu "Aide" est sélectionnée.
6. **OuvertureMenuParam**
  - État lorsque le menu "Paramètres" est cliqué.
7. **EnleverDiapo**
  - État lorsque le bouton "EnleverDiapo" est cliqué pour retirer le diaporama courant.
8. **AfficheDiapo**

- État où le diaporama est affiché.
- 9. **ImagePrécédente**
  - État de l'application lorsque l'utilisateur clique sur le bouton "Précédent".
- 10. **ImageSuivante**
  - État de l'application lorsque l'utilisateur clique sur le bouton "Suivant".
- 11. **ChangementVitesseDefilement**
  - État lorsque l'option "Vitesse Defilement" est cliquée dans le menu "Paramètres" et la vitesse de défilement est modifiée.
- 12. **Lancement du mode Automatique**
  - État lorsque le mode automatique du diaporama est lancé.
- 13. **Arrêter le diaporama**
  - État où le diaporama automatique est arrêté.
- 14. **FermetureProg**
  - État de l'application lorsque l'utilisateur clique sur le bouton de fermeture.
- 15. **Final**
  - Représenté par le cercle avec un point au centre (fin du processus).
  - Indique que l'application s'est fermée.

## Transitions

1. **Lancement de l'application**
  - Transition de l'état initial vers AfficheLecteurDiapo.
2. **Menu "Fichier" cliqué**
  - Transition de AfficheLecteurDiapo vers OuvertureMenuFichier.
3. **Bouton "ChargerLeDiaporama" cliqué**
  - Transition de OuvertureMenuFichier vers AfficheDiapo.
4. **Menu "Aide" cliqué**
  - Transition de AfficheLecteurDiapo vers OuvertureMenuAide.
5. **Fichier "AproposDe" cliqué**
  - Transition de OuvertureMenuAide vers OuvertureAproposDe.
6. **Menu "Paramètres" cliqué**
  - Transition de AfficheLecteurDiapo vers OuvertureMenuParam.
7. **Bouton "EnleverDiapo" cliqué**
  - Transition de AfficheLecteurDiapo vers EnleverDiapo.
8. **Bouton "Vitesse Defilement" cliqué**
  - Transition de OuvertureMenuParam vers ChangementVitesseDefilement.
9. **Changement de vitesse de défilement**
  - Transition de ChangementVitesseDefilement vers AfficheDiapo.
10. **Bouton "Précédent" cliqué**
  - Transition de AfficheDiapo vers ImagePrécédente.
11. **Bouton "Suivant" cliqué**
  - Transition de AfficheDiapo vers ImageSuivante.
12. **Bouton de fermeture cliqué**
  - Transition de AfficheLecteurDiapo vers FermetureProg.

- Transition de ImagePrécédente vers FermetureProg.
- Transition de ImageSuivante vers FermetureProg.

### 13. Arrêter le diaporama cliqué

- Transition de Lancement du mode Automatique vers Arrêter le diaporama.

## Synthèse

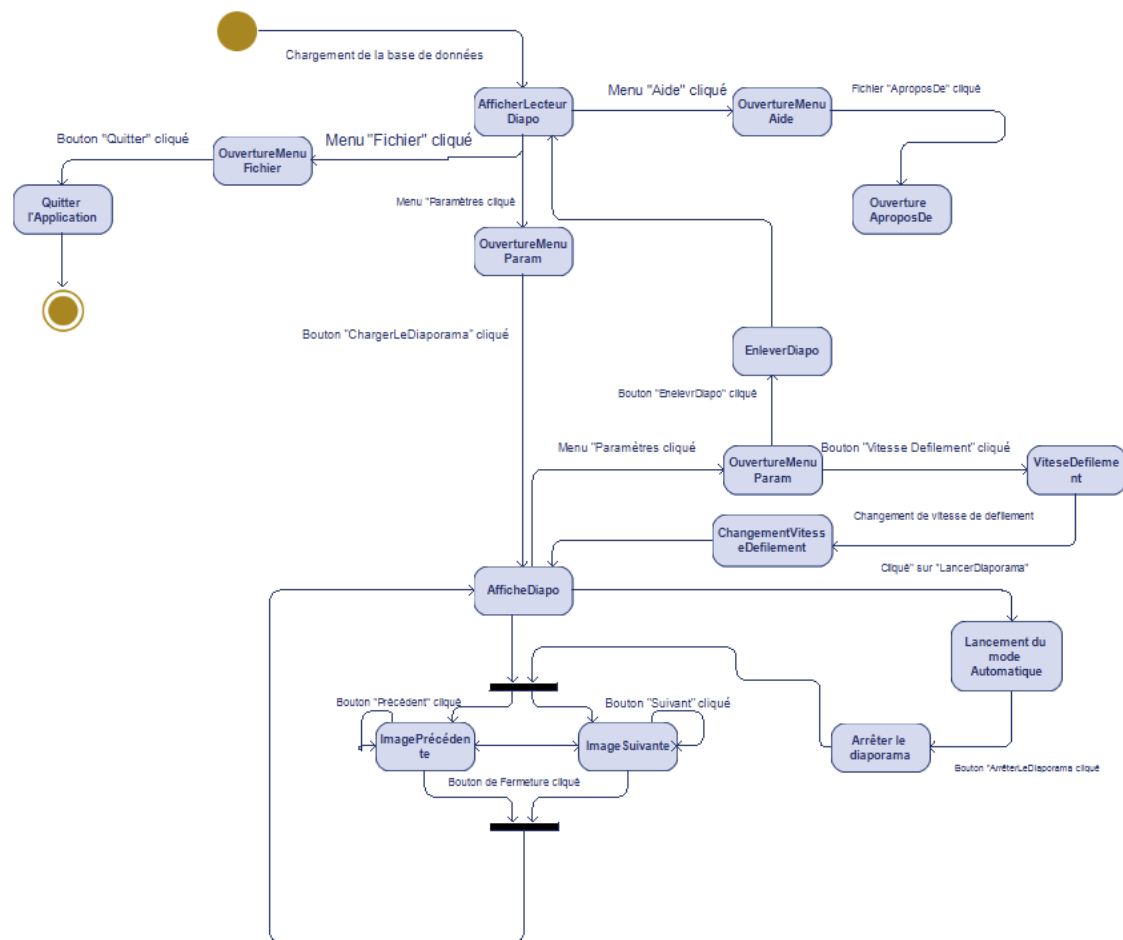
Ce diagramme montre les différents chemins que peut prendre l'application de diaporama en fonction des actions de l'utilisateur. Il inclut des transitions pour charger et enlever des diaporamas, afficher des informations d'aide, ajuster les paramètres tels que la vitesse de défilement, naviguer entre les images, et fermer l'application. Le flux est plus complexe que le premier diagramme, couvrant une gamme plus large de fonctionnalités utilisateur.

## V6 :

### Rappel de la version :

v6 / v6_MVP	<p>Projet créé par duplication de la v5 (ou v5_MVP)</p> <ul style="list-style-type: none"><li>- Chargement des informations relatives aux images à partir de la <b>Base de Données</b></li></ul> <p><b>A faire pour parvenir à cette version :</b></p> <p>Dossier de conception :</p> <ul style="list-style-type: none"><li>- Mise à jour de la description de la version</li></ul> <p>Code :</p> <ul style="list-style-type: none"><li>- Mise en œuvre de la fonctionnalité annoncée</li></ul>
-------------------	---

## Diagramme état-transition :



L'état-transition est le même que à la V5 sauf que on connecte la base de données à l'état initial.

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

Événement \ État	Initial	OuvertureMenu Fichier	AfficheLecteur Diapo	OuvertureMenu Aide	OuvertureApropos De	OuvertureMenu Param	EnleverDiapo	AfficheDiapo	ImagePrécédente	ImageSuivante	Changement VitesseDefile ment	Lancement du mode Automatique	Arrêter le diaporama	FermetureProg	Final
Lancement de l'application	X		X												
Menu "Fichier" cliqué		X													
Bouton "ChargerLeDiap orama" cliqué								X							
Menu "Aide" cliqué			X												
Fichier "AproposDe" cliqué				X											
Menu "Paramètres" cliqué						X									
Bouton "EnleverDiapo" cliqué							X								
Bouton "Vitesse Defilement" cliqué											X				
Changement de vitesse de défilement								X							
Bouton "Précédent" cliqué									X						
Bouton "Suivant" cliqué										X					
Bouton de fermeture cliqué			X						X	X				X	
Arrêter le diaporama cliqué												X			
Fermeture de l'application							Pour le 09/06/2024							X	X

## Bilan:

### Ce que nous avons appris :

Ce projet nous a offert l'opportunité d'apprendre à gérer l'affichage et les transitions d'états de manière approfondie, ainsi qu'à établir des connexions avec des bases de données. Grâce à ces compétences, nous avons pu comprendre et maîtriser la structuration d'un projet de manière modulaire et efficace. Ces acquis nous ont permis d'appréhender comment organiser chaque composant de manière cohérente, de faciliter la maintenance et l'évolution du projet, et d'assurer une intégration fluide entre les différentes parties du système. En somme, cette SAE a renforcé notre capacité à concevoir des projets bien structurés et fonctionnels.

### Ce que nous avons aimé / pas aimé :

Nous avons apprécié ce type de projet car il offre une expérience concrète de développement, nous permettant de voir directement l'application de nos compétences. Cependant, nous avons rencontré des défis, notamment en ce qui concerne la répartition du temps et les fonctionnalités limitées du lecteur de diaporama par défaut. La contrainte de temps a été particulièrement frustrante, de même que la répartition des fonctionnalités entre les différentes versions, ce qui a parfois entraîné des délais et des attentes désynchronisés.

### Ce qui a été difficile :

La partie la plus difficile a été la conception initiale du lecteur. Il était crucial de partir sur de bonnes bases, mais cela a pris beaucoup de temps et a parfois ralenti notre progression. De plus, l'implication des membres du groupe n'a pas toujours été optimale, ce qui a posé des défis supplémentaires.

Comprendre les attentes du projet a également été un point difficile, tout comme l'accompagnement obligatoire et non volontaire de la part des étudiants, qui a parfois manqué de clarté et de soutien. Ces éléments ont contribué à des périodes de confusion et de ralentissement dans le développement du projet.

Le temps passé sur le projet :

	<b>ETCHECOPAR Maxime</b>	<b>LAGÜE Théo</b>	<b>PIGNEAUX Loris</b>
<b>Conception code</b>	3h	10h	3h
<b>Documentation livrable</b>	5h	1h30	5h
<b>Livrable</b>	2h30	0h	3h
<b>Temps total</b>	10h30	11h30	11h



