# 

**Développement d'une application**

S2.01

*TD1 – TP2*

*Cazalaa Emile - Davaud Zélie - Mohtar Lamina*

*2024-2025 /BUT informatique 1ére année , deuxième semestre*

url Github : https://github.com/ecazalaa/SAE2.01

# 

**Table des matières :**

[**v1 3**](#_6s3xz9rq27l)

[Diagramme de classe : 3](#_l4e0udb82ub7)

[**Image 4**](#_8loo2ebafcfa)

[**ImageDansDiapo 5**](#_jm0ealypt7ru)

[**Diaporama 6**](#_bwafdp88728b)

[**Lecteur 7**](#_8x5rarsm32dq)

[Relations et cardinalités 9](#_x1niq3htgr9h)

[**v2 10**](#_9zgx9duivpxz)

[Diagramme d’Etat : 10](#_1pqdyepylh4x)

[Liens entre éléments d’interface et fonctionnalités : 11](#_gqirhnby8xkk)

[**Diagramme de classe : 12**](#_rs6yz17s7wp)

[**Classes rajoutée : 13**](#_gzt8u78p2gvo)

[LecteurVue 13](#_h55xq9tcti06)

[Presentation 13](#_8vgwetjcx233)

[**Mvp 3: 15**](#_jo5xpbh7ed79)

[Mise a jour de la version : 15](#_uord5mdl2ell)

[Modifications dans les tables : 15](#_d9i9ioxcuyod)

[Méthodes publiques ajoutées : 15](#_lr4cyunj3e2q)

[Lecteur 16](#_iiswy3yhup58)

[Attributs privés ajoutés : 16](#_gtx4qcorqjlp)

[Méthodes publiques supprimées : 16](#_ddbes6m3lqft)

[Méthodes publiques ajoutées : 17](#_ar0ice8al7og)

[Méthodes privées ajoutées : 17](#_fxx51yhg6k5o)

[**Mvp 4: 17**](#_jn0qvljpt934)

[Mise a jour de la version : 17](#_493t6r6py7u2)

[**Mvp 6: 19**](#_7l7i47mcpyp)

[Diagramme de classe : 19](#_g2xajycotm2d)

[Diagramme d’Etat : 20](#_u9uwd7wge1x8)

[**Classes rajoutée : 21**](#_gm9vz479falu)

[DataBase 21](#_ncmrwudbjty9)

[**Mvp 7: 21**](#_ilviu28ws9bt)

[Mise a jour de la version : 21](#_1ta242tyx2j)

[**Bilan: 22**](#_o6xjcon6ocz9)

[- Ce que vous avez appris ? 22](#_krotoi9rqwj1)

[- Ce que vous avez aimé / pas aimé ? 22](#_nhnpqbskw6sz)

[- Ce qui a été difficile? 22](#_sduf5xoz2ak5)

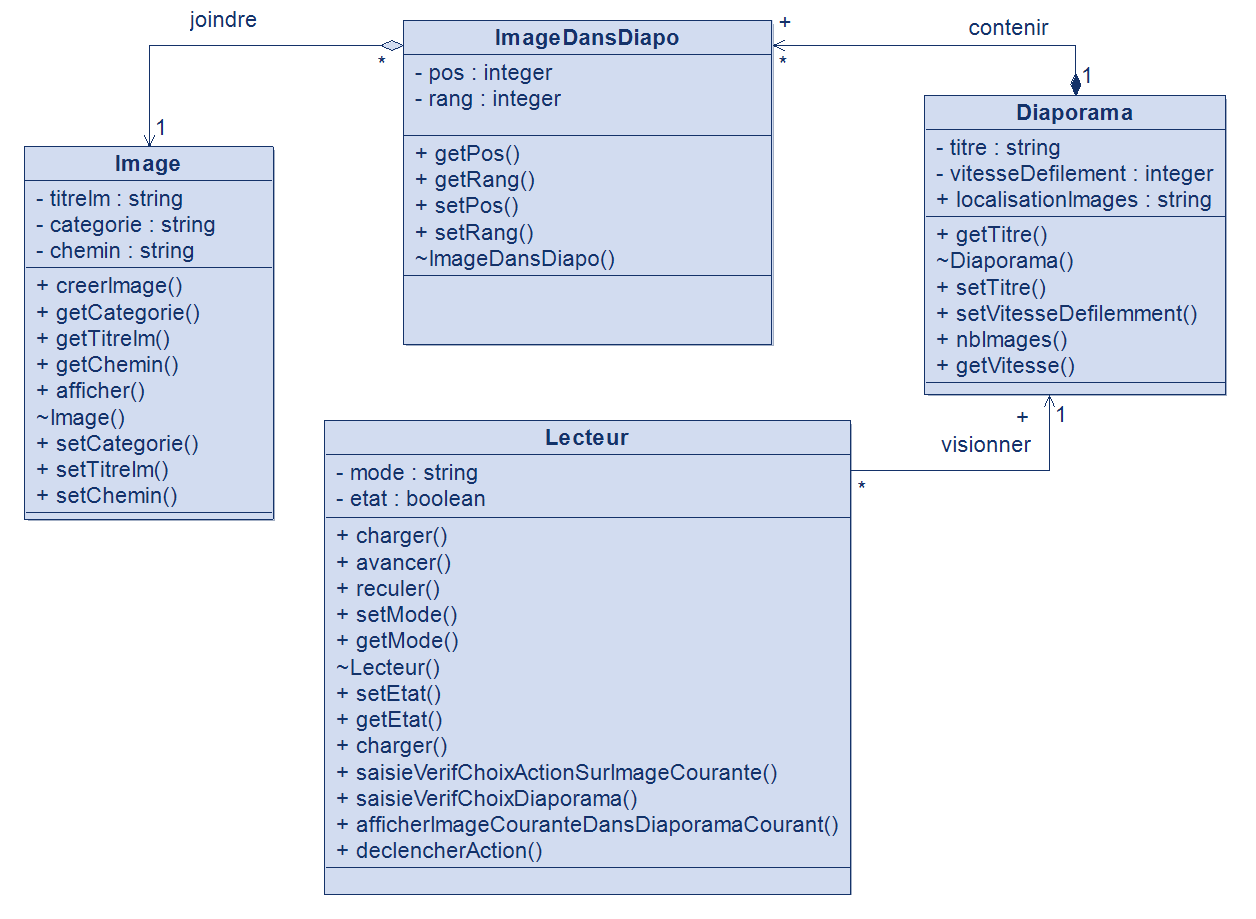
[- Le temps passé (sur conception/sur code) ? 23](#_qp3t347acmp2)

[- Ce que vous auriez pu faire mieux ? 23](#_hp54llmqnb2e)

[- Ce qui pourrait être amélioré dans la saé? 23](#_lvadkoncp0ej)

# **v1**

## **Diagramme de classe :**



## **Image**

Attributs

| **Nom de l’attribut** | **Signification** | **But** |
| --- | --- | --- |
| \_titreIm | Titre de l’image | Défini le titre à une image |
| \_categorie | catégorie de l’image | représente la catégorie à laquelle l'image appartient. |
| \_chemin | chemin de l’image | représente le chemin d'accès de l’image |

Méthodes

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| creerImage() | créer une image | Permet d'instancier des nouvelles images à partir des attributs entrés en paramètres |
| getCategorie() | obtenir la catégorie | Guetteurs permettant de récupérer la catégorie de l’image |
| setCategorie() | definir la catégorie | Setteur , la méthode va définir la catégorie afin de l'associer à l’image à partir du paramètre pris en entré |
| getTitrelm() | obtenir le titre de l’image | Guetteur , la méthode va récupérer le titre associé à l’image |
| setTitrelm() | definir le titre de l’image | Setteur permettant de définir le titre associé à l’image à partir du paramètre pris en entrée |
| getChemin() | obtenir le chemin de l’image | Guetteur qui permet de récupérer le chemin associé à l’image |
| setChemin() | définir le chemin de l’image | Setteur qui va définir le chemin  associer à l’image à partir du paramètre pris en entrée |
| afficher() | afficher l’image | Ici , la méthode va afficher l’image |
| ~Image() | destructeur de la classe Image | Libérer les ressources associées à “Image” avant sa destruction. |

# 

## **ImageDansDiapo**

Attributs

| **Nom de l’attribut** | **Signification** | **But** |
| --- | --- | --- |
| \_pos | position de l'image dans le tableau d'images | Ordre de chargement initial des images dans la table des images |
| \_rang | rang de l'image dans le diaporama | Défini le rang de l’image dans le diaporama |

Méthodes

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| setPos() | definir la position de l’image | Setteur, va définir la position de l’imageDansDiapo associée a partir du paramètre pris en entré |
| getPos() | renvoie la position de l’image | Guetteur , la méthode va récupérer la position de l'imageDansDiapo associée |
| setRang() | definir le rang de l’image | Setteur, va définir le rang de l’imageDansDiapo associée à partir du paramètre pris en entrée |
| getRang() | renvoie le rang de l’image | Guetteur, la méthode va récupérer le rang de l'imageDansDiapo associée |
| ~ImageDansDiaporama() | destructeur de la classe ImageDansDiaporama | libérer les ressources associées à “ImageDansDiaporama” avant sa destruction. |

## **Diaporama**

Attributs

| **Nom de l’attribut** | **Signification** | **But** |
| --- | --- | --- |
| \_titre | titre du diaporama | Cet attribut représente le titre ou l'intitulé du diaporama. |
| \_vitesseDefilement | vitesse de défilement du diaporama | Cet attribut représente la vitesse de défilement des images du diaporama. |
| \_localisationImages | tableau dynamique permettant la localisation des images dans le diapo | Ce vecteur (ou tableau dynamique) contenant des éléments de type ImageDansDiapo permet de localiser les ImagesDansDiapo dans le diaporama |

Méthodes

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| setTitre() | definir le titre du diaporama | Setteur, va définir le titre du diapo associé à partir du paramètre pris en entrée |
| getTitre() | renvoie le titre du diaporama | Getteur, la méthode va renvoyer le titre du diaporama |
| setVitesseDefilement() | définir la vitesse de défilement des images du diaporama | Setteur qui permet à l'utilisateur de modifier la vitesse de défilement selon ses préférences. |
| getVitesse() | récupérer la vitesse de défilement des images du diaporama | Guetteur, permettre à l'utilisateur de récupérer la vitesse de défilement |
| NbImages() | renvoie le nombre d’images du diaporama | Permet de récupérer le nombre d’images du diaporama |
| ~Diaporama() | destructeur | Libérer les ressources associées à “Diaporama “ avant sa destruction |

## **Lecteur**

Attributs

| **Nom de l’attribut** | **Signification** | **But** |
| --- | --- | --- |
| \_mode | permet de connaître le mode du lecteur . | Connaître le mode du lecteur (il peut être manuel ou automatique) |
| \_etat | boolean qui permet de connaître l'état du lecteur | Savoir si le lecteur est vide ou chargé |

Méthodes

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| ~Lecteur() | destructeur | Libérer toutes les ressources allouées à l'objet de la classe avant sa destruction |
| charger() | Chargement du tableau des diaporamas avec des diaporamas préalablement construits 'en dur' | Permet de charger le tableau des diaporamas avec un diaporama entré en paramètre |
| charger() | Chargement du tableau des images avec seules les informations textuelles de quelques images. | Permet de charger le tableau des images avec une image entrée en paramètre |
| avancer() | avancer le diaporama | Méthode qui permet d’avancer le diaporama en cours |
| reculer() | reculer le diaporama | Méthode qui permet de reculer le diaporama en cours |
| setEtat() | permet de spécifier l'état du lecteur de diaporama | Setteur pour définir l'état associé au Lecteur à partir du paramètre pris en entré |
| getEtat() | permet de récupérer l'état de fonctionnement du lecteur de diaporama | Guetteur permettant de récupérer l'état du Lecteur (vide ou chargé) |
| setMode() | permet de spécifier le mode de fonctionnement du lecteur de diaporama | Setteur qui fournit une interface pour modifier le mode de fonctionnement du lecteur de diaporama selon les besoins de l'utilisateur. En permettant à l'utilisateur de choisir entre le mode automatique et le mode manuel |
| getMode() | permet de récupérer le mode de fonctionnement du lecteur de diaporama | Getteur. La méthode va récupérer le mode utilisé par l’utilisateur. |
| saisieVerifChoixActionSurImageCourante() | Saisie du choix d'action de l'utilisateur vis-à-vis de l'image et diaporama courants | Procédure permettant à l’utilisateur d’indiquer une action correcte à effectuer. |
| saisieVerifChoixDiaporama() | Saisie par l’utilisateur du diaporama souhaité jusqu’à ce qu’il soit possible/correct. | Retourne un numéro de diaporama choisi par l'utilisateur (dans la liste des diaporamas existants) |
| afficherImageCouranteDansDiaporamaCourant() | Affichage à l'écran des infos de l'image courante dans son diaporama | On pourra récupérer des informations comme :  - le titre du diaporama  - L’image courante : titre, famille, contenu  - Le rang de l'image courante dans le diaporama / nbre total d'images du diaporama |
| declencherAction() | L’utilisateur peut choisir l’action qu’il souhaite réaliser. | Selon le pChoix fait par l'utilisateur, réalise une des actions suivantes: A)vancer, R)eculer, C)hoisir un autre diaporama, Q)quitter |

## **Relations et cardinalités**

## 

**Relations entre les classes :**

Pour la réalisation de ce projet notre équipe a décidé de créer 4 classes differentes , ces classes sont : Image , ImageDansDiapo ,Diaporama et Lecteur .

Parmi ces 4 classes, on retrouve 3 associations .

Tout d’abord on retrouve l’association de **joindre** entre la classe “ Image” et la classe “Image Dans Diapo". Une Image peut joindre un seul diaporama et de l'autre côté la classe ImageDansDiapo peut être jointe par plusieurs images.

On retrouve entre ces deux classes une association **d'agrégation** , en effet l'objet dépendant reste dans la portée d'une relation même lorsque l'objet source est détruit.

Ensuite, on retrouve l’association de **contenir** entre la classe

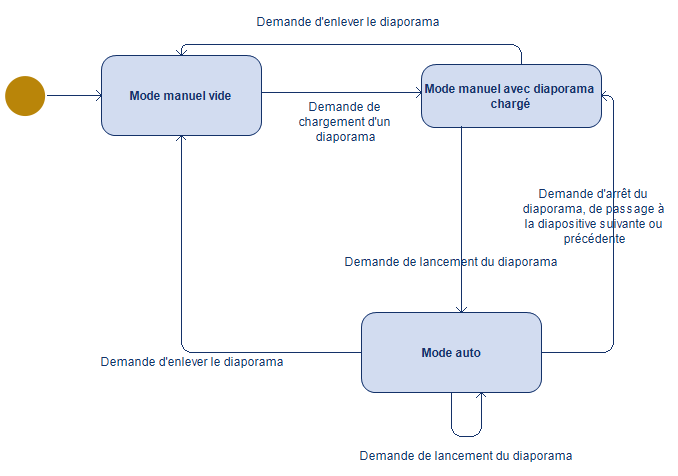
“Image Dans Diapo” et la classe “Diaporama”. Un diaporama peut contenir une seule et unique image et de l’autre côté une Image peut être contenu par plusieurs Diaporamas.

On retrouve entre ces deux classes une association de **composition** . Cela veut dire que si la classe diaporama est effacée ,les objets contenus dans la classe ImagedansDiapo le seront aussi .

Pour finir, on retrouve l’association de **visionner** entre la classe “Diaporama” et “Lecteur”. Un diaporama peut être visionné par un seul lecteur et d'un autre côté un lecteur peut visionner plusieurs diaporamas .

# **v2**

## **Diagramme d’Etat :**



| Éléments d’interface déclencheurs des événements→ | acCharger | acEnlever | btnArret | btnSuivant | btnPrecedent | btnLancer |
| --- | --- | --- | --- | --- | --- | --- |
| Événements→  Etats ↓ | Demande de chargement d’un diaporama | Demande d’enlever le diaporama | Demande d’arrêt du diaporama | Demande de passage à la diapositive suivante | Demande de passage à la diapositive précédente | Demande de lancement du diaporama |
| *Mode manuel vide*  *(état 1)* | état 2 | … | … | … | … | … |
| Mode manuel avec diaporama chargé  (état 2) | … | état 1 | … | … | … | état 3 |
| Mode auto  (état 3) | … | état 1 | état 2 | état 2 | état 2 | état 3 |

## **Liens entre éléments d’interface et fonctionnalités :**

acCharger : action du menu paramètres pour charger un diaporama.

acEnlever : action du menu paramètre permettant de retirer un diaporaama précédemment chargé.

acQuitter : action du menu Fichier permettant de quitter l’application.

acAProposDe : action du menu Aide, permettant de voir la version de l’application et ses auteurs.

labMode : label affichant le mode de fonctionnement en cours.

btnSuivant : Permet en mode manuel d’afficher la diapositive suivante et en mode auto de repasser en mode manuel.

btnPrecedent : Permet en mode manuel d’afficher la diapositive précédente et en mode auto de repasser en mode manuel.

btnLancer : Permet le lancement du diaporama en mode auto depuis le début.

btnArret : Permet de repasser en mode manuel, depuis le mode auto, inactif en mode manuel.

labTitre : label affichant le titre de la diapositive.

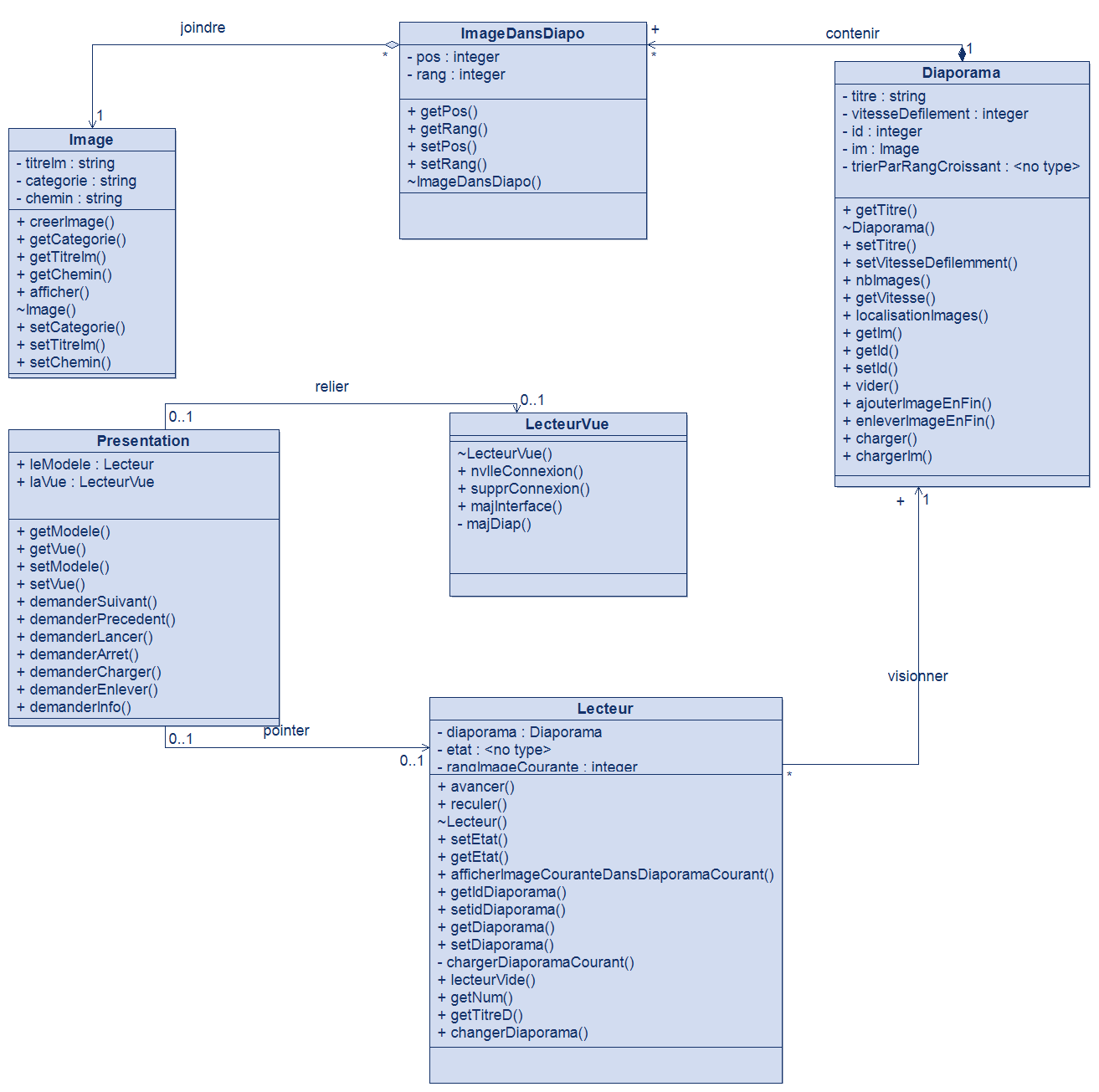
labCat : label affichant la catégorie de l’image.

labDiapo : label affichant l’image de la diapositive.

labNum : label affichant le numéro de la diapositive.

# 

# **Diagramme de classe :**



# 

# **Classes rajoutée :**

## **LecteurVue**

Méthodes

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| nvlleConnexion  () | créer une connexion avec la présentation | méthode crée une nouvelle connexion avec le lecteur |
| supprConnexion  () | supprimer la connexion avec la présentation | méthode qui va venir supprimer la connexion avec le lecteur |
| majInterface  () | Mise à jour de l’interface | méthode qui va venir faire une mise à jour de l’interface du lecteur |

## 

## **Presentation**

Attributs

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| \_leModele | pointeur sur le modèle | donner accès à toutes les méthodes liée au modèle |
| \_laVue | pointeur sur la vue | avoir a accès à la vu pour lui permettre de changer l’interface |

Méthodes

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| demanderSuivant() | demande de passer à la diapo suivante | cette méthode va permettre de passer au diapo suivant de la présentation |
| demanderPrecedent() | demande de passer à la diapo precedente | cette méthode va permettre de passer au diapo précédent de la présentation |
| demanderLancer() | demande de lancer la présentation | cette méthode va permettre de lancer la présentation |
| demanderArret() | demande d'arrêter la présentation | cette méthode va permettre de demander l'arrêt de la présentation |
| demanderCharger( | demander de charger la présentation | cette méthode va permettre de demander le chargement de la présentation |
| demanderEnlever() | demande d’enlever la présentation | cette méthode va permettre de demander d’enlever la présentation |
| demanderInfo() | demande l’information concernant la présentation | cette méthode va permettre de demander l’information concernant la présentation |

# 

# **Mvp 3:**

## **Mise a jour de la version :**

L'objectif principal de cette version est d'intégrer certaines fonctionnalités de la version 1 tout en affichant des images dans l'interface visuelle.

- Pas de base de données : toujours création des images ‘en dur’

- Affichage des images dans l’interface graphique

- Fonctionnement du lecteur en mode Manuel uniquement

- Le menu Fichier >> Quitter permet d’arrêter l’application

- Le menu Aide >> A propos de… ouvre une Boîte de Message donnant des informations sur :

- la version de l’application

- la date de création

- les auteurs

## **Modifications dans les tables :**

On trouve des modifications ( ajouts / suppression) sur :

**Diaporama**

### **Méthodes publiques ajoutées :**

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| trierParRangCroissant() | trie les images du diaporama par ordre de rang croissant | méthode qui va va venir trie les images du diaporama par ordre de rang croissant |
| getIm() | recupere l’image | getteur qui va permettre de récupérer une image |
| getId() | récupère l’identifiant du diaporama | getteur qui va permettre de récupérer l’identifiant du diaporama |
| setId() | permet de spécifier l’identifiant du diaporama | setteur qui va permettre de spécifier l’identifiant du diaporama |
| vider() | vider le diaporama | méthode qui va permettre de vider le diaporama |
| ajouterImageEnFin() | ajouter une image au diaporama | méthode qui va permettre d’ajouter une image au diaporama (en fin de l'attribut images) |
| enleverImageEnFin() | enlève une image au diaporama | méthode qui va permettre d’enlever une image du diaporama (en fin de l'attribut images) |
| charger() | chargement du diaporama courant | méthode qui va permettre le chargement, dans l'attribut images, des images associées au diaporama courant (d'attribut id) |
| chargerIm() | chargement des images dans le diapo courant | méthode qui va permettre de charger les images associé au diaporama |

### **Lecteur**

### **Attributs privés ajoutés :**

* unsigned int RangImageCourante;
* unsigned int idDiaporama;

### **Méthodes publiques supprimées :**

* void charger(Diaporamas& diaporama);
* void charger(Images& pImages);
* void avancer(const Diaporama& pDiaporama, unsigned int& pPosImageCourante);
* void reculer(const Diaporama& pDiaporama, unsigned int& pPosImageCourante);
* void saisieVerifChoixActionSurImageCourante(char& pChoixAction);
* unsigned int saisieVerifChoixDiaporama(const Diaporamas& pDiaporamas);
* void declencherAction(char pChoixAction, const Diaporamas& pDiaporamas, unsigned int& pDiaporamaCourant, unsigned int& pImageCourante, const Images& pImages);

### 

### **Méthodes publiques ajoutées :**

*On ajoute 15 méthodes de plus au total , qui sont :*

* unsigned int getIdDiaporama() const;
* Diaporama\* getDiaporama() const;
* unsigned int getRangImageCourante() const;
* string getImageCourante() const;
* string getCatImageCourante() const;
* string getChemImageCourante() const;
* unsigned int nbImages() const;
* bool lecteurVide() const
* void setDiaporama(Diaporama\* pDiaporama);
* void setRangImageCourante(unsigned int pRImageCourante);
* void setIdDiaporama(unsigned int pIdDiaporama);
* void changerDiaporama(unsigned int pId);
* string getTitreD();
* string getNum();
* void viderLecteur();

**LecteurVue**

#### **Méthodes privées ajoutées :**

* majDiap(string num, string tD, string tI, string tC, string chem);
* afficheQMessageBox(); // Affiche une fenêtre de dialogue avec les informations concernant l’application .

# 

# **Mvp 4:**

## **Mise a jour de la version :**

L'objectif principal de cette version est de mettre en oeuvre le mode “Auto” par duplication de la v3 mvp

**Mvp 5:**

**Mise a jour de la version :**

Projet créé par duplication de la v4 (ou v4\_MVP)

- Implémentation de Fichier >> Paramétrer >> Vitesse … : ouvre une Boîte de Dialogue permettant de modifier la vitesse de défilement des images

- Au démarrage, aucun Diaporama n’est chargé

- Implémentation de Fichier >> Paramétrer >> Charger diaporama … : qui charge les images ‘en dur’ et les mets à disposition triées par ordre croissant de rang

- Implémentation de Fichier >> Paramétrer >> Enlever diaporama … : qui enlève le Diaporama en cours. Le lecteur est vide à nouveau

**Information :**

On avait déjà implémenter Charger et Enlever diaporama dans les versions précédentes donc a du juste implémenter la Vitesse.

# 

# 

# **Mvp 6:**

## **Diagramme de classe :**

## 

## 

## 

## 

## 

## 

## **Diagramme d’Etat :**

| Éléments d’interface déclencheurs des événements→ | acAProposDe | acVitesse | acCharger | acEnlever | btnArret | btnSuivant | btnPrecedent | btnLancer |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Événements→  Etats ↓ | Demande l’information du Lecteur | Demande de changer la vitesse d’un diaporama | Demande de chargement d’un diaporama | Demande d’enlever le diaporama | Demande d’arrêt  du diaporama | Demande de passage à la diapositive suivante | Demande de passage à la diapositive précédente | Demande de lancement du diaporama |
| *Mode suivant(1)* | *5* | … | 4 | … | … | … | 2 | … |
| Mode précédent(2) | 5 | … | 4 |  | … | 1 | … | 9 |
| Mode automatique(3) | 5 | 8 | 4 |  |  | 9 | 9 | … |
| Mode charge(4) | 5 | 8 | … | 7 | … | 1 | 2 | 9 |
| Mode info (5) | … | … | 4 |  |  | 1 | 2 | 9 |
| Mode lancer(6) | 5 | … | 4 | 4 | 4 | 1 | 2 | 9 |
| Mode enlever(7) | 5 | 8 | 4 | … | 4 | 1 | 2 |  |
| changer la vitesse(8) | 5 | … | 4 | 7 | 4 | 1 | 2 |  |
| Mode manuel (9) | 5 | 8 | 4 | 7 |  | 1 | 2 | 3 |

## 

# 

# **Classes rajoutée :**

## **DataBase**

Attributs

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| mydb | base de données | pouvoir travailler sur la base de données |

Méthodes

| **Nom de la méthode** | **Signification** | **But** |
| --- | --- | --- |
| openDataBase() | boolean pour ouvrir la base de données | méthode qui nous permet d’ouvrir ou pas la base de données |
| closeDataBas () | fermer la base de données | méthode qui nous permet de fermer notre base de doonées |

# **Mvp 7:**

## **Mise a jour de la version :**

Dans cette version créé par duplication de la v6 (ou v6\_MVP)

On trouve un :

- Chargement des diaporamas à partir de la base de données

- Nouvelle implémentation de Fichier >> Paramétrer >> Charger diaporama … :

qui charge les informations relatives au Diaporama et aux Images à partir de la Bases de

Données

# **Bilan:**

## 

## 

## - Ce que vous avez appris ?

On a appris à faire des versions mvp ,travailler avec les base de données, insérer des images, incorporer nos travails et surtout se familiariser avec Qt , outil qu’on connaissait pas avant.

## 

## - Ce que vous avez aimé / pas aimé ?

* : On a aimé pouvoir travailler avec une intervalle pour la première fois , on est vraiment rentrés dans le rôle de “développeur” .
* : point négatif , le temps . On aurait aimé que certaines versions soient rassemblées, ce qui nous aurait donner plus de temps .

## 

## - Ce qui a été difficile?

Ce que nous avons trouvé difficile dans ce projet a été de nous organiser en tant qu'équipe afin de terminer toutes les tâches à temps.

Le temps a été un vrai obstacle au début car on eu peur de ne pas pouvoir tout rendre.

Au début, il était aussi difficile d'incorporer les versions mvp car nous ne l'avions jamais fait auparavant, mais peu à peu la tâche est devenue plus facile, ce qui nous a permis de faire toutes les versions en mvp.

## 

## - Le temps passé (sur conception/sur code) ?

| temps réaliser : | Zelie | Lamina | Emile |
| --- | --- | --- | --- |
| conception | 2H | 16H | 2H |
| code | 28H | 8H | 22H |
| temps personnel par rapport au temps de la SAE | 30/19.5H | 24/19.5H | 24/19.5H |

Pour le **code** on prend en compte la gestion d’erreurs , les heures hors cours,toutes les versions , les implémentations, etc …

Pour la **conception** on prend en compte la réalisation des classes , des diagrammes , la réalisation des tableau , du rapport , du bilan , de la mise à jour en Git … etc

## - Ce que vous auriez pu faire mieux ?

On aurait pu faire la v8 avec un peu plus de temps et optimiser le choix des sous-programmes (regrouper et trier ).

De plus, nous aurions pu passer moins de temps sur la documentation en améliorant la qualité de nos fichiers .h.

## - Ce qui pourrait être amélioré dans la saé?

On aurait aimé avoir un peu plus de séances avec des professeurs afin de pouvoir poser plus de questions .