BUT Informatique Semestre 2 2023-2024

IUT de Bayonne et du Pays Basque, Département Informatique

S2.01 « *Développement d’une application* »

CLEMENCEAU Edouard, TD 2 - TP 4

MASSON Rafael, TD 2 - TP 4

VINET LATRILLE Jules, TD 2 - TP 4

1ere Année de formation

Lien GitHub : https://github.com/jvlatrille/LecteurDiaporama.git

**Sommaire.**

Table des matières

[Version 1 : Projet non graphique Orienté Objets 3](#__RefHeading___Toc714_153876483)

[Diagramme des classes UML 3](#__RefHeading___Toc716_153876483)

[Attributs et méthodes de chaque classe 4](#__RefHeading___Toc718_153876483)

[Image 4](#__RefHeading___Toc720_153876483)

[Version 2 : Projet non graphique Orienté Objets 8](#__RefHeading___Toc722_153876483)

[Bilan 8](#__RefHeading___Toc724_153876483)

[Ce que l’on a appris 8](#__RefHeading___Toc726_153876483)

[Ce qu’on a aimé / pas aimé 8](#__RefHeading___Toc728_153876483)

[Ce qui a été difficile 8](#__RefHeading___Toc730_153876483)

[Le temps passé (sur conception / code) 8](#__RefHeading___Toc732_153876483)

[Ce que vous auriez pu faire de mieux (avec le recul) 9](#__RefHeading___Toc734_153876483)

[Ce qui pourrait être amélioré dans la SAE 9](#__RefHeading___Toc736_153876483)

# **Version 1 : Projet non graphique Orienté Objets**

## Diagramme des classes UML



## Attributs et méthodes de chaque classe

### Image

|  |  |  |
| --- | --- | --- |
| Attributs | | |
| Nom | Type | Signification |
| m\_titre | Chaîne de caractères | Titre de l’image |
| m\_categorie | Chaîne de caractères | Catégorie de l’image |
| m\_chemin | Chaîne de caractères | Chemin de l’image |

|  |  |
| --- | --- |
| Méthodes | |
| Nom | Signification et but |
| Image() | Constructeur par défaut |
| getTitre() | Getter qui permet de récupérerle titre de l'image |
| getChemin() | Getter qui permet de récupérerle chemin de l'image |
| getCategorie() | Getter qui permet de récupérerla catégorie de l'image |
| setTitre() | Setter qui permet de définir le titre de l'image |
| setChemin() | Setter qui permet de définir le chemin de l'image |
| setCategorie() | Setter qui permet de définir la catégorie de l'image |
| afficher() | Afficher tous les détails de l'image |

### ImageDansDiapo

|  |  |  |
| --- | --- | --- |
| Attributs | | |
| Nom | Type | Signification |
| m\_rang | Entier non signé | Rang de l'image dans le diaporama |
| m\_pos | Entier non signé | Position de l'image dans le diaporama |

|  |  |  |
| --- | --- | --- |
| Méthodes | | |
| Nom | Signification et but | |
| ImageDansDiaporama() | Constructeur par défaut | Créer une instance d'image dans un diaporama avec des valeurs par défaut |
| getRang() | Getter qui permet de récupérerle rang de l'image | |
| getPos() | Getter qui permet de récupérerla position de l'image | |
| setRang() | Setter qui permet de définir le rang de l'image | |
| setPos() | Setter qui permet de définir la position de l'image | |
| afficherImageCourante() | Afficher les détails (position et rang) de l'image dans le diaporama | |

### Diaporama

|  |  |  |
| --- | --- | --- |
| Attributs | | |
| Nom | Type | Signification |
| m\_titre | Chaîne de caractères | Titre du diaporama |
| m\_vitesseDefilement | Entier non signé | Vitesse de défilement du diaporama |
| m\_posImageCourante | Entier non signé | Position de l'image courante dans le diaporama |

|  |  |  |
| --- | --- | --- |
| Méthodes | | |
| Nom | Signification et but | |
| Diaporama() | Constructeur par défaut | Créer une instance de diaporama avec des valeurs par défau |
| getTitre() | Getter qui permet de récupérerle titre du diaporama | |
| getVitesseDefilement() | Getter qui permet de récupérerla vitesse de défilement du diaporama | |
| getNombreImages() | Getter qui permet de récupérerle nombre d'images dans le diaporama | |
| getPosImageCourante() | Getter qui permet de récupérerla position de l'image courante dans le diaporama | |
| setTitre() | Setter qui permet de définir le titre du diaporama | |
| setVitesseDefilement() | Setter qui permet de définir la vitesse de défilement du diaporama | |
| setNombreImages() | Setter qui permet de définir le nombre d'images dans le diaporama | |
| setPosImageCourante() | Setter qui permet de définir la position de l'image courante dans le diaporama | |
| ajouterImage() | Ajouter une image dans le diaporama | |
| avancer() | Avancer vers l'image suivante dans le diaporama | |
| reculer() | Reculer vers l'image précédente dans le diaporama | |
| afficherImageCouranteDansDiaporamaCourant() | Afficher les détails de l'image courante dans le diaporama | |
| triCroissantRang() | Trier les images du diaporama par leur rang | |

### Lecteur

|  |  |  |
| --- | --- | --- |
| Attributs | | |
| Nom | Type | Signification |
| m\_numDiapoCourant | Entier non signé | Numéro du diaporama courant |

|  |  |  |
| --- | --- | --- |
| Méthodes | | |
| Nom | Signification et but | |
| Lecteur() | Constructeur par défaut | Créer une instance de lecteur de diaporamas avec des valeurs par défaut |
| getNumDiapoCourant() | Getter qui permet de récupérerle numéro du diaporama courant | |
| getNombreDiapos() | Getter qui permet de récupérerle nombre total de diaporamas | |
| setAllDiapos() | Setter qui permet de définir toutes les diapositives | |
| setNumDiapoCourant() | Setter qui permet de définir le numéro du diaporama courant | |
| setNombreDiapos() | Setter qui permet de définir le nombre total de diaporamas | |
| declencherAction() | Déclencher une action dans le lecteur | |
| saisieChoixActionImageCourante() | Saisir le choix d'action pour l'image courante | |
| saisieChoixDiaporama() | Saisir le choix de diaporama | |
| chargerImage() | Charger les images | |
| chargerDiapos() | Charger les diaporamas | |
| afficherDiapoCourant() | Afficher le diaporama courant | |

# **Version 2 : Projet non graphique Orienté Objets**

## Diagramme état-transition de l’application sous ses 2 formes

### Diagramme état-transition

Une image contenant texte, capture d’écran, diagramme, Parallèle

Description générée automatiquement

### Version matricielle

## Documentation des liens entre éléments d’interface et fonctionnalités

### Boutons (QPushBouton) :

Précédent : Permet de retourner à l’image précédente lorsque l’on est en mode de défilement manuel.

Pause : Permet de mettre le défilement en pause lorsqu’on est en mode de défilement automatique.

Suivant : Permet de passer à l’image suivante lorsque l’on est en mode de défilement manuel.

### Actions de la barre de menu (QMenuBar, QMenu et QAction) :

Fichier > Quitter : Permet de fermer l’application.

Fichier > Charger diaporama : Permet de charger un nouveau diaporama (qu’il y en ait un déjà chargé ou pas).

Paramètres > Enlever diaporama : Permet d’enlever le diaporama qui est chargé.

Paramètres > Vitesse de défilement : Permet d’ouvrir une boite de dialogue qui donne la possibilité de changer la vitesse de défilement des images en défilement automatique.

Mode > Automatique : Permet de changer de mode de défilement pour passer en mode automatique . Par défaut le mode est en manuel.

Mode > Manuel : Permet de changer de mode de défilement pour passer en mode manuel. Par défaut le mode est en manuel.

Aide > A propos : Permet d’afficher la version actuelle du lecteur de diaporama et ses auteurs.

# **Bilan**

## Ce que l’on a appris

Nous avons appris à mieux comprendre un code qui nous était inconnu et à réaliser des classes à partir de ce code. Nous avons aussi perfectionné notre usage de Git/GitHub pour la sauvegarde de fichiers.

## Ce qu’on a aimé / pas aimé

Dans cette version, nous avons apprécié la mise en place de GitHub (bien que périlleuse dans un premier temps). Aussi, la réflexion pour créer le diagramme de classe était agréable (nous avons une bonne cohésion de groupe).

Nous n'avons pas apprécié le fait de repasser à plusieurs le code pour trouver les erreurs dans les nombreux fichiers, et faire en sorte de l’optimiser et qu’il fonctionne au mieux (même si nous sommes conscient que c’est normal / nécessaire si nous souhaitons rendre un travail de qualité).

## Ce qui a été difficile

L’assimilation du code a été périlleuse, mais le plus compliquer était trouver toutes les méthodes. Nous avons dû repasser plusieurs fois par-dessus pour être sûr de ne pas en oublier, ce qui nous a pris du temps.

## Le temps passé (sur conception / code)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | CLEMENCEAU Edouard | | MASSON Rafael | | VINET LATRILLE Jules | |
| Versions | Conception | Code | Conception | Code | Conception | Code |
| Version 1 | 3h | 4h | 3h | 4h30 | 2h | 6h30 |
| Version 2 | 2h | 1h | 2h | 1h | 2h | 1h |
| Version 3 |  |  |  |  |  |  |
| Version 4 |  |  |  |  |  |  |
| Version 5 |  |  |  |  |  |  |

## Ce que vous auriez pu faire de mieux (avec le recul)

Nous aurions certainement pu trouver plus de méthodes qui nous simplifieraient la vie dans le futur développement du lecteur de diaporama ou qui le rendraient plus optimisé. Cependant, nous pensons que nous n'avons pas encore assez de recul pour en parler avec certitude.

## Ce qui pourrait être amélioré dans la SAE

Ce qui pourrait être amélioré dans la SAE serait d'avoir plus de cours encadrés pour pouvoir poser davantage de questions aux professeurs, ou bien d'allouer plus d'heures globales (encadrées / en autonomie) afin de permettre aux étudiants de perfectionner leur travail.

Aussi, habituer les étudiants à GIT dès le début de la première année serait une bonne idée. Ce n’est pas compliqué à utiliser, et ça leur permettrait de mieux organiser leur travail.