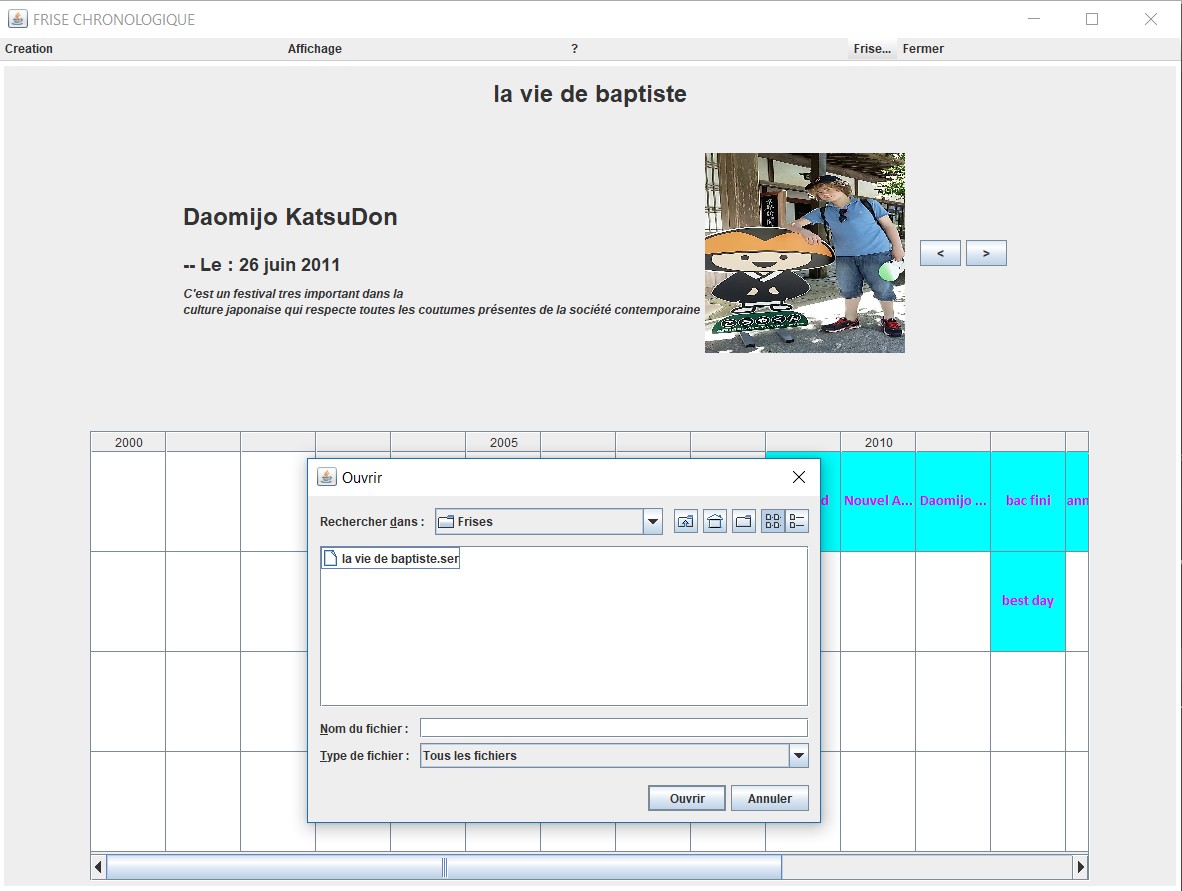
**RAPPORT DU PROJET TUTEURÉ**

GUIDE DE SURVIE DU VOYAGEUR TEMPOREL



Sommaire

Contenu

[1- INTRODUCTION 3](#_Toc516135464)

***2-CONCEPTION GÉNÉRALE***

***3-CONCEPTION DÉTAILLÉE***

[PACKAGE MODELE 7](#_Toc516135465)

[PACKAGE CONTROLEUR 8](#_Toc516135466)

[PACKAGE VUE 8](#_Toc516135467)

[Conclusion 13](#_Toc516135468)

***4- MANUEL D'UTILISATION***

[0- Avant de commencer 14](#_Toc516135469)

[1- Démarrage de la machine temporelle 14](#_Toc516135470)

[A- Il y a déjà un fichier qui existe dans le dossier 14](#_Toc516135471)

[B- Aucune frise n’existe dans le dossier ! 14](#_Toc516135472)

[2- Les différentes fenêtres et la navigation 15](#_Toc516135473)

[A- La création d’évènement 15](#_Toc516135474)

[B- La Frise Chronologique 16](#_Toc516135475)

[C- Ouvrir et Créer une frise 16](#_Toc516135476)

[D- Quitter l’application 17](#_Toc516135477)

# INTRODUCTION

Voici le Rapport Complet du projet réalisé par Aussenac Baptiste et Castello Nicolas pour le second semestre. Vous trouverez ici le déroulement du projet, avec le dossier de conception générale, détaillée, les différents tests mis en œuvre ainsi qu’une notice d’utilisation de l’application

1. CONCEPTION GÉNÉRALE

L'utilisateur a le choix entre plusieurs options de la fenêtre mère : il peut **créer une frise ou un événement**, **ouvrir une frise existante**, **afficher la frise et un événement** ou **quitter**.

Pour pouvoir ouvrir ou afficher une frise il faut d'abord l’avoir créée. De plus, pour afficher un événement il faut aussi l’avoir créé avant, ce qui implique de créer une frise dans un premier temps comme le montre ce diagramme de cas d’utilisation ci-dessous :

.

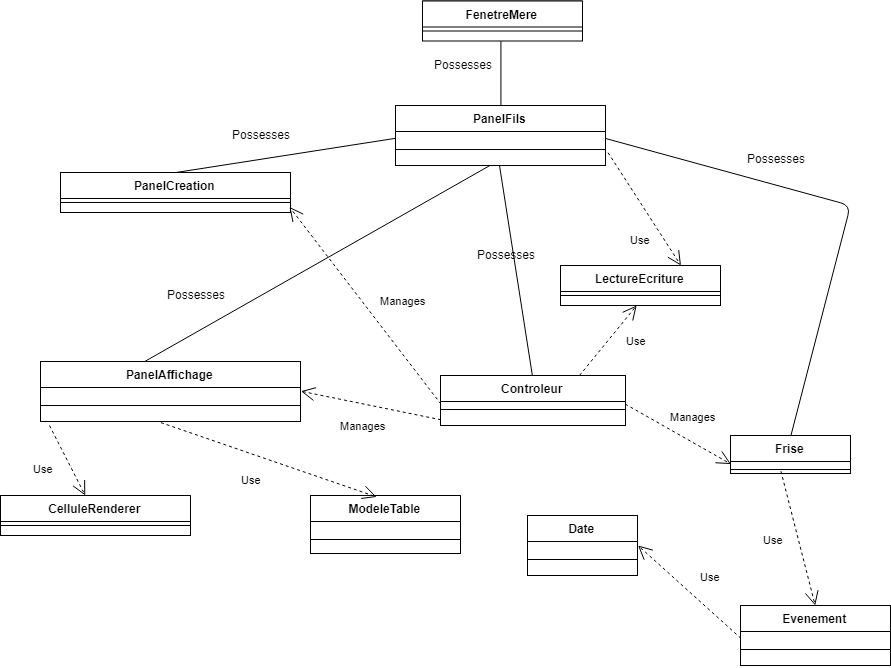
L'application utilise plusieurs classes qui interagissent entre elles. Ces classes sont reparties dans différents packages :

* modèle qui contient les classes Date, Evènement, Frise et LectureEcriture.
* vue qui contient les classes CelluleRenderer, FenetreMere, ModeleTable, PanelAffichage, PanelCreation, PanelFils.
* controleur qui contient la classe Controleur.

La classe FenetreMere possède un PanelFils qui contient une Frise, un PanelCreation, un PanelAffichage et un Controleur.

Le Controleur et le PanelFils utilisent LectureEcriture.  
Le Controleur gère la Frise, le PanelAffichage et le PanelCreation.   
La Frise utilise la classe Evènement qui utilise la classe Date. Le PanelAffichage utilise les classes CelluleRenderer et ModeleTable.

Voici un diagramme de classe à haut niveau d’abstraction pour représenter nos dires



Avec le diagramme de séquences nous pouvons voir dans le temps comment les classes interagissent entre elles.

S'il n'y a pas de frise l'utilisateur va en créer une dans PanelFils sinon une frise s’affiche dans PanelFils.



Quand l'utilisateur va sur le formulaire le PanelFils affiche le PanelCreation, l'utilisateur remplit les cases du formulaire puis PanelCreation envoie les informations au Controleur qui crée l’événement.

Si l'utilisateur va sur l'affichage de la frise, le PanelFils affiche le PanelAffichage, puis l'utilisateur peut sélectionner les événements qu'il veut voir sur PanelAffichage.

L'utilisateur peut créer une nouvelle frise ou ouvrir une frise dans le PanelFils qui changera la frise courante par la nouvelle.

Le PanelFils peut permettre à l'utilisateur de quitter le programme.



1. CONCEPTION DÉTAILLÉE

Pour ce projet, nous avons décidé de partir sur une architecture MVC comme vu dans les cours

# PACKAGE MODELE

Ce package contient toute les objets utile à l'application,

* une classe Date qui permet d’avoir un jour, un mois et une année.



* Une classe Evènement qui utilise une date et possède un titre, un poids, une description et un nom d'image.



* La classe Frise qui utilise des évènements trier par année par valeur du poids, une année de début et de fin ainsi qu'un intituler



* Pour finir la classe LectureEcriture qui permet d'écrire et de lire des fichiers.



# PACKAGE CONTROLEUR

Ce package contient un Controleur qui permet de synchroniser les classes de modele et de vue.

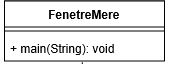
* La classe Controleur utilise un PanelCreation, un PanelAffichage et une Frise. Si l'utilisateur clic sur le bouton pour choisir une image on parcoure les dossiers présent, une foi une image choisie le chemin est enregistrer dans un label. S’il clic sur le bouton ajout le Controleur passe les donner saisie dans le PanelCreation a la Frise et écrit dans le fichier de la frise, il met ensuit a jour le PanelAffichage et réinitialise le PanelCreation. Si on clique sur les boutons '<' et '>' du PanelAffichage on affiche l'évènement précèdent ou suivant dans le PanelAffichage.



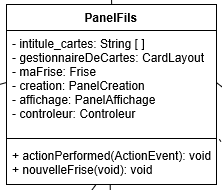
# PACKAGE VUE

Ce package permet de visualiser les classes du modele pour l'utilisateur, c'est l'interface de l'application.

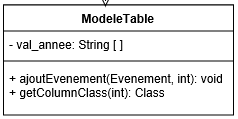
* La classe FenetreMere contient la barre de menu et un PanelFils.



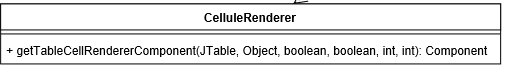
* La classe PanelFils contient PanelAfichage et PanelCreation ainsi qu'un Controleur et une Frise. Si on sélectionne 'Creation' dans le menu PanelFils affiche le PanelCreation, et pour 'Affichage' il prend le PanelAffichage, si on sélection le '?' il ouvre une fenêtre de dialogue, la sélection de 'ouvrir une frise' ouvre la sélection des fichiers pour récupérer une frise existante et la transmettre aux autres champs, 'Nouvelle frise' ouvre des fenêtres de dialogue pour entrer les différents paramètres d'une frise et 'Quitter' ferme l'application.



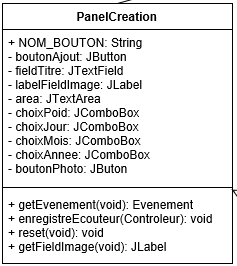
* La classe ModeleTable indique la structure de la JTable que nous évoquerons plus tard. C'est elle qui affichera les événements de la frise sur la JTable.



* La classe CelluleRenderer permet de changer la couleur de la case et du texte du tableau pour chaque événement.



* La classe PanelCreation utilise un GridBagConstraints pour la disposition des diffèrent éléments du formulaire, PanelCreation peut récupérer une date, un label, un évènement et réinitialiser le formulaire.



* La classe PanelAffichage est séparé en trois parties :

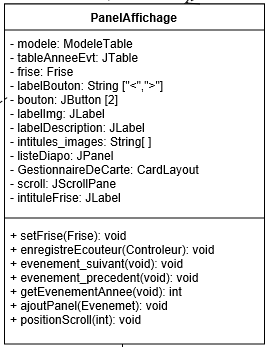
Nous avons donc utilisé pour cela un BorderLayout, pour afficher ces trois parties en haut, au centre et au sud.

Dans un premier temps il y a l'intitulé de la frise qui prend la partie nord, pour définir dans quelle frise chronologique nous nous situons.

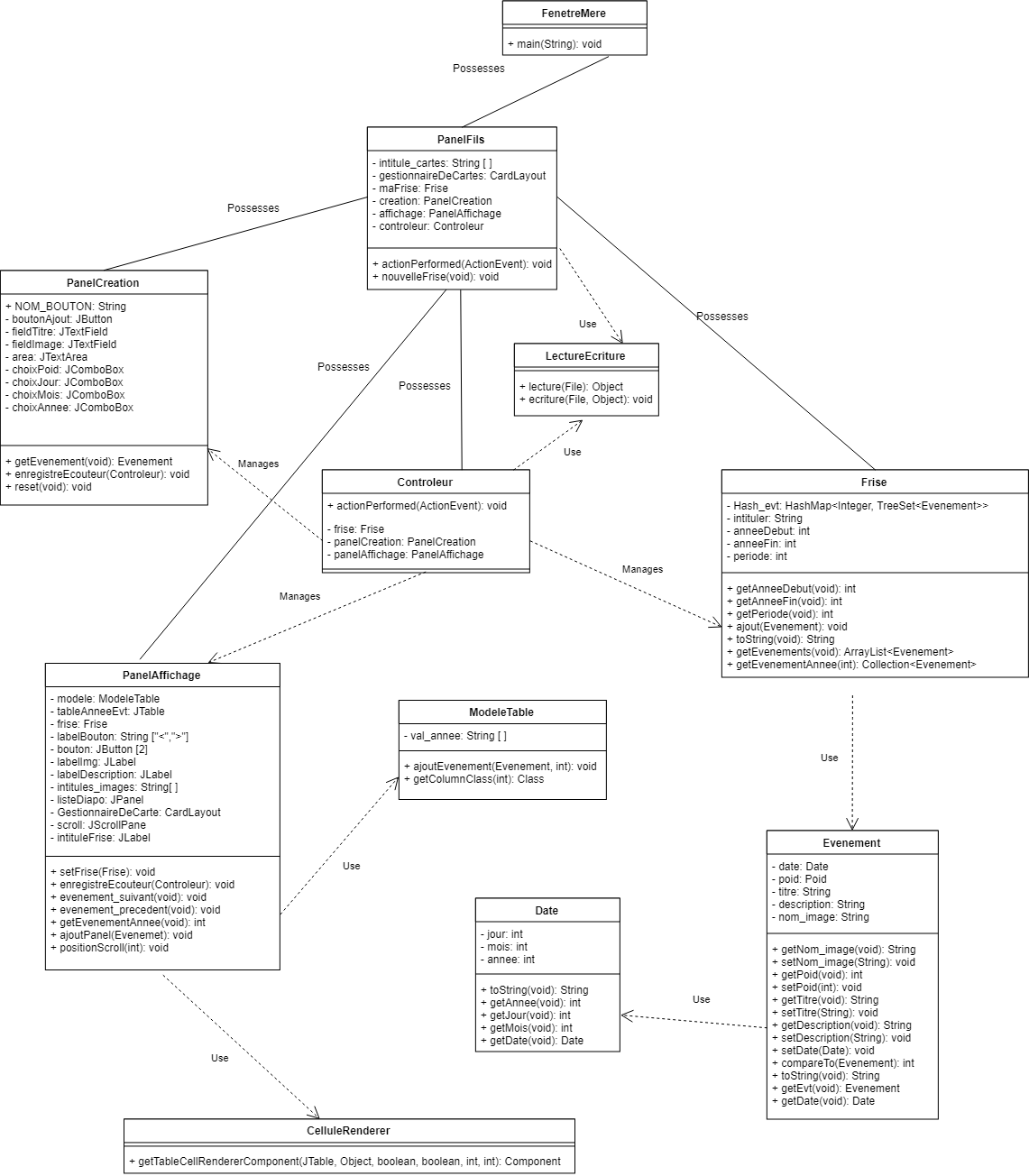
Ensuite, il a une JTable qui affichera les évènements du HashMap de la Frise. Pour cela. Nous avons donc le ModeleTable qui les ajoutera dans la table. De plus, afin d'avoir plus de clarté, et pour séparer les cellules qui contiennent des événements des autres, nous avons créé la classe CelluleRenderer qui est abordé ci-dessus.

Pour finir, il y a la partie centrale de l'affichage. C'est le diaporama d'évènements. Constitué d'un CardLayout, le diaporama permet d'afficher tous les évènements de la frise chronologique, avec l'image qui lui a été associée.

Lors de chaque ajout d'évènements, le PanelAffichage ajoute ce dernier dans le diaporama. Par ailleurs, deux boutons permettent de naviguer entre les évènements précédant et suivant l'évènement qui est affiché sur le diaporama.(cette gestion d’action se fait bien entendu dans le Controleur)  
Lorsque ces boutons sont utilisés, la table se synchronise en affichant dans la même volée la position de l'évènement qui est demandé (suivant ou précédent).



Ce qui nous donne, en conclusion, ce diagramme



Pas très lisible certes… il reprend le diagramme vu dans la conception générale avec en plus de cela, toutes les méthodes et les champs vu dans les différentes classes

# Conclusion

Afin de conclure les documents de conceptions, voici quelques améliorations possibles que nous pourrions établir sur le projet :

* Afficher les photos sur le tableau en modifiant le Renderer,
* Modifier et Supprimer un évènement,
* Rendre l’application plus esthétique.
* Gérer le cas où, lors du démarrage de l’application et qu’il n’y a pas de frise dans le dossier « Frises », qu’il demande à l’utilisateur si il veut en créer une ou ouvrir une déjà existante.

Lors de la réalisation de l’application, nous avons eu quelques difficultés sur la synchronisation entre la scrollPane et le diaporama, qui n’est pas tout-à-fait corrigé.

De plus, l’ajout de la recherche de fichiers à travers l’ordinateur a aussi été quelque peu troublant car c’était un domaine encore non exploité lors de la pratique en cours.

1. MANUEL D’UTILISATION

GUIDE DE SURVIE DU VOYAGEUR TEMPOREL

# Avant de commencer

Notez qu’il est important de préciser un point avant de lancer l’application.

* Les frises qui sont écrites sont gérés dans le Fichier « Frises » et sont enregistrés sous le format « .ser » il faut donc le **CRÉER** si ce n’est pas déjà fait.
* Si l’on ouvre une frise dans un dossier autre que le «  Frises », alors une copie s’effectuera dans le dossier « Frises ».

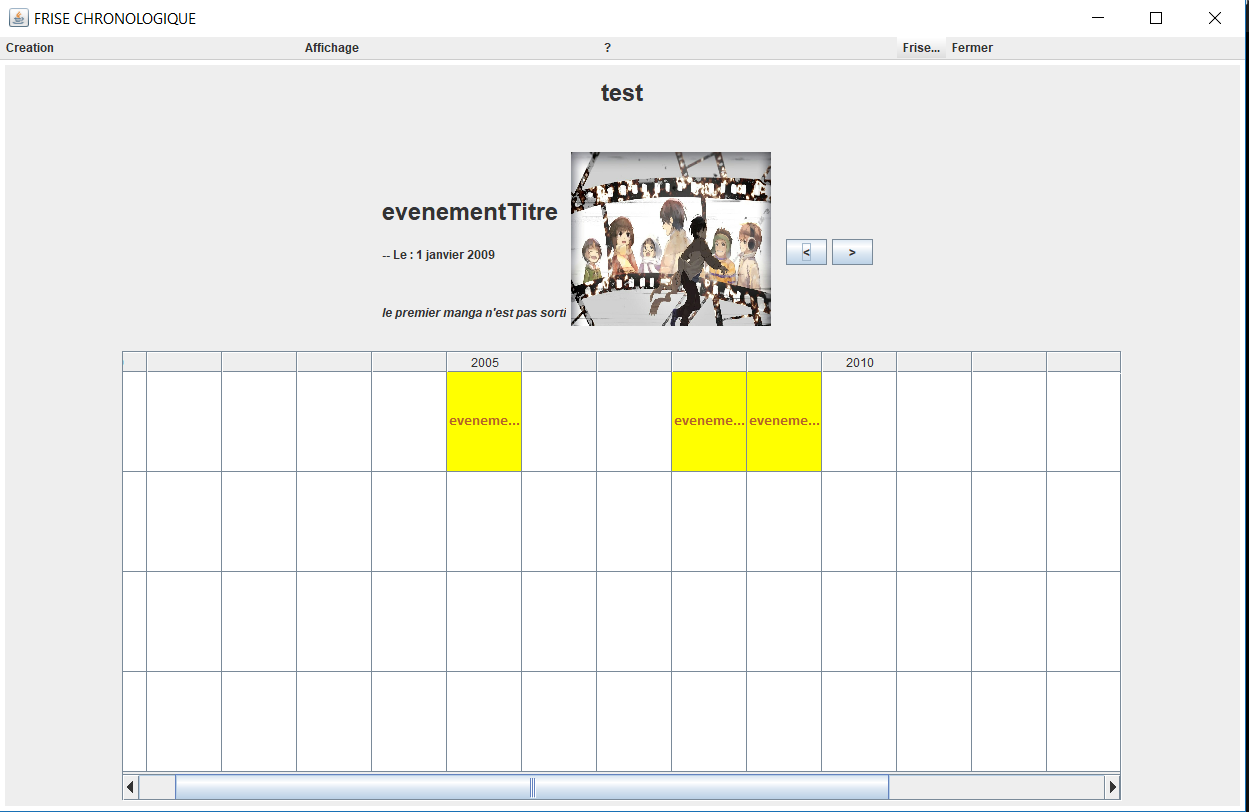
Un évènement à forcément un poids, si deux évènements ont le même poids à la même année, alors le dernier évènement créé ne sera pas affiché sur la table

# Démarrage de la machine temporelle

Il y a deux choix possibles lors du lancement de l’application.

## Il y a déjà un fichier qui existe dans le dossier

L’application lancera la première frise situé dans le dossier « Frises » (dans l’ordre alphabétique).

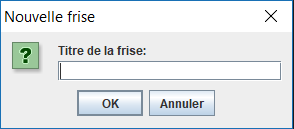


## Aucune frise n’existe dans le dossier !

S’il n’y a aucune frise alors le programme va ouvrir plusieurs Pop-up qui vont permettre à l’utilisateur de créer une nouvelle frise qui sera donc affiché juste ensuite.

Il est demandé :

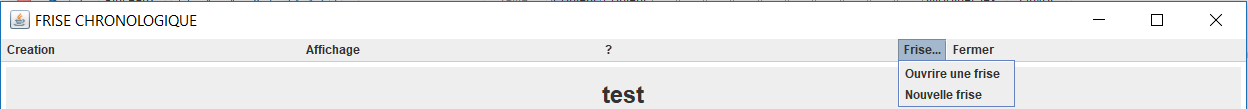
* Le Titre de la Frise Chronologique
* L’année de début de la frise
* L’année de fin de la frise
* La période d’affichage des années (utile pour le tableau).



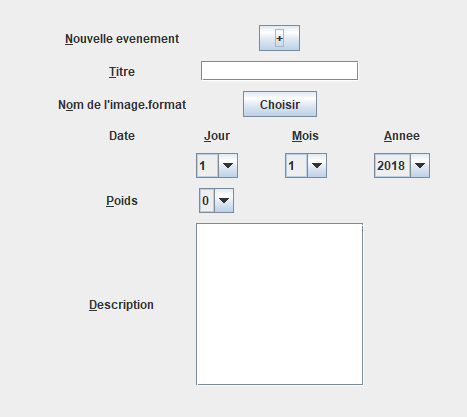
Attention ! : Si une frise chronologique existe déjà mais qu’elle n’est pas dans le dossier « Frises », alors il faut la déplacer dans le Dossier.

# Les différentes fenêtres et la navigation

Un voyageur du temps doit se permettre de naviguer librement entre les différents menus qui lui sont proposés. Il y a donc un menu mis à disposition pour effectuer les différentes fonctionnalités



## La création d’évènement

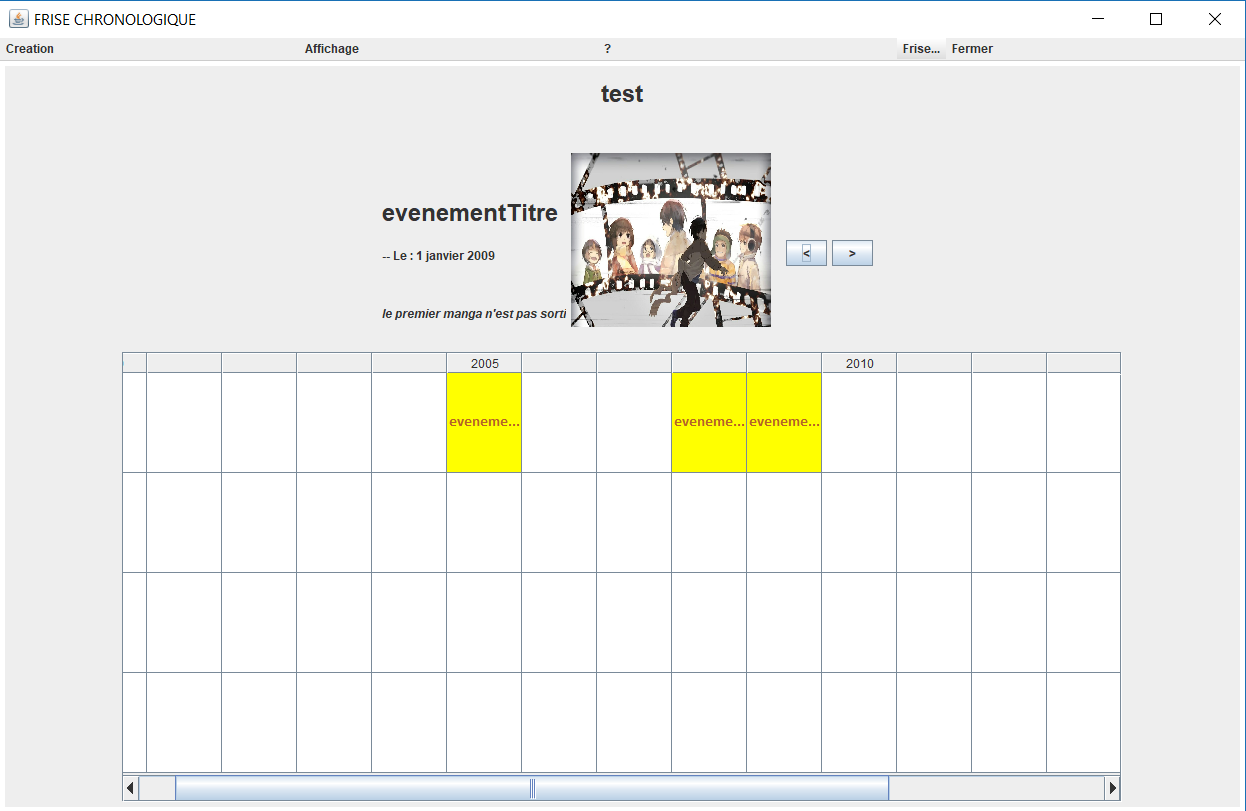
Pour créer des évènements dans la frise, il faut remplir le formulaire suivant : 

Il est possible de remplir chaque champs avec la souris ou en utilisant la Mnémonique.

Si aucune image n’est sélectionnée, alors il n’y aura pas de photo affiché sur le diaporama

Pour ajouter l’évènement dans la Frise Chronologique, il suffit d’appuyer sur le bouton  ou d’appuyer sur ALT+ N.

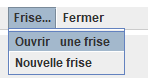
## La Frise Chronologique



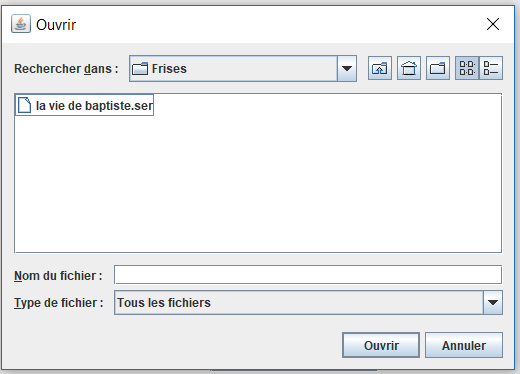
On peut naviguer entre les différents évènements de la frise grâce aux flèches « < » et «  > ». Si l’on clique sur les cases où il y a un évènement, alors il s’affichera aussi en haut de la fenêtre.

## Ouvrir et Créer une frise

1. Ouvrir



Lorsqu’on clique sur « ouvrir une frise », une petite fenêtre demande alors la localisation du fichier



1. Créer

Créer la frise reprend la même situation lors du démarrage (cf : [Aucune frise n’existe dans le dossier !](#_Aucune_frise_n’existe))

## Quitter l’application

Deux façons de quitter l’application sont possibles :

* Fermer avec la Croix rouge en haut à droite
* Fermer en cliquant sur l’icône « fermer » dans le menu qui redemandera l’accord de l’utilisateur.