Java ayant des capacités limitées concernant la lecture de fichiers multimédias, nous nous sommes tournés vers une librairie externe pour accomplir cette tâche. Nous avons choisi la librairie **vlcj** qui fournit des bindings vers la librairie native « libvlc ». Cette libraire permet de décoder la plupart des formats audio et vidéo sans avoir à se soucier d’installer des codecs.

Le lecteur vidéo utilise **Swing** pour l’affichage de la vidéo et des contrôles. La raison pour laquelle nous utilisons **Swing** et non **JavaFx** est que pour bénéficier de l’accélération graphique il est nécessaire d’utiliser des composants **AWT** « heavyweight » dont l’équivalent n’est pas disponible en **JavaFx**. Après une première tentative d’implémentation en **JavaFx**, on a bien pu vérifier qu’il fallait utiliser l’accélération graphique, car il n’était pas possible de lire une vidéo de manière fluide.

A la base le lecteur vidéo offrait un « overlay » pour le contrôle de la vidéo. Pour réaliser une telle interface, il fallait créer une deuxième fenêtre superposée au lecteur vidéo. Cette fenêtre doit avoir une position synchronisée avec celle du lecteur. Le problème avec cette méthode est qu’au moment de passer le lecteur vidéo en plein écran, il arrivait que le gestionnaire de fenêtre du système d’exploitations décide de mettre l’ « overlay » soit en arrière-plan, soit en plein écran. Pour résoudre ce problème, nous avons intégré le menu de contrôle directement dans la fenêtre principale du lecteur.

La gestion de la synchronisation s’effectue d’une manière similaire à celle du lecteur audio.

**Remarque**: Un composant « heavyweight » est un composant qui utilise l’API graphique fournie par le système d’exploitation.

Le lecteur vidéo se repose sur deux classes princiaples : la classe « Player » et la classe « ControlPanel ».

La classe « Player » est un « singleton » fournissant des méthodes permettant de lire une vidéo et la mettre en pause. Il y a également des méthodes permettant d’avancer ou de reculer la vidéo de deux secondes ainsi que de position le lecteur à un temps donné. Chacune de ces méthodes intègre également la possibilité d’envoyer des événements utilisés par le gestionnaire de synchronisation. Le « ControlPanel » est un élément **Swing** contrôlant directement le lecteur au moyen des méthodes mentionnées précédemment.