

# Rapport: Projet Anamorphose

ALOS Mickaël, COLLOT Kévin, GUILBAULT Maxime, HOUDAYER Corentin

14-05-2018

# Table des matières

<b>1</b>	<b>Présentation du sujet</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	Compréhension du sujet . . . . .	2
<b>2</b>	<b>Etude théorique</b>	<b>3</b>
2.1	Exigences . . . . .	3
2.2	Diagrammes UML . . . . .	4
2.2.1	Cas d'utilisation . . . . .	4
2.2.2	Diagramme d'activité . . . . .	5
2.3	Problèmes et solutions proposées . . . . .	6

# Chapitre 1

## Présentation du sujet

### 1.1 Introduction

L'objectif de ce projet est de créer une application mobile permettant de simuler et de reproduire le phénomène d'anamorphose, connu en photographie, pour créer une image anamorphosée à partir d'une vidéo.

### 1.2 Compréhension du sujet

L'anamorphose est un phénomène qui peut se produire lorsque l'on tente de prendre en photo un objet se déplaçant à une vitesse relative élevée par rapport à celle des rideaux de l'obturateur de l'appareil photo. La photographie alors obtenue peut présenter des déformations sur les parties mobiles du sujet photographié. A travers ce sujet, notre but est donc de simuler le mouvement des rideaux sur une vidéo où la caméra est immobile et les sujets filmés se déplacent. En plus du mouvement vertical des rideaux présents dans les appareils photographiques, nous pourrions implanter d'autres mouvements (horizontaux, en diagonal ou personnalisés par l'utilisateur).

# Chapitre 2

## Etude théorique

### 2.1 Exigences

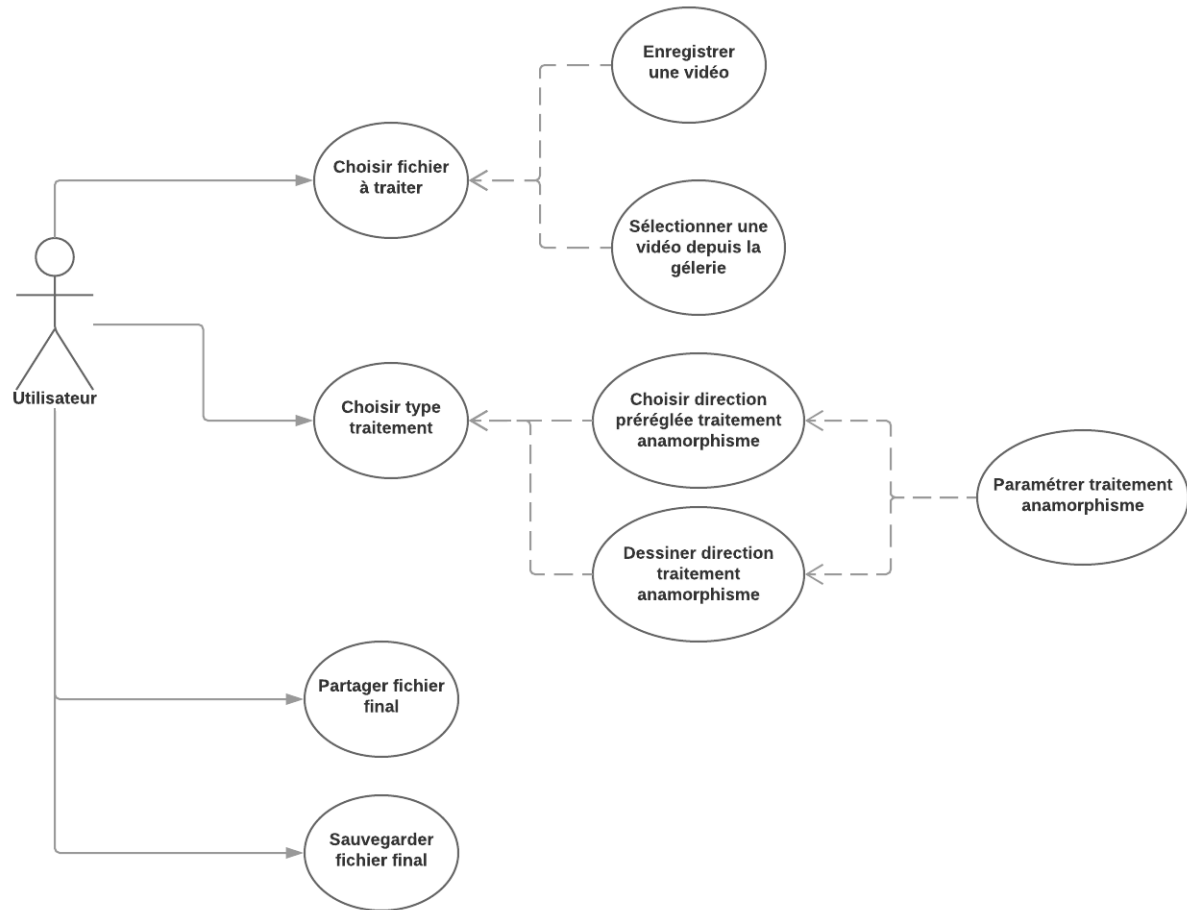
Depuis le sujet fourni, nous avons dégagés les exigences suivantes pour notre application finale :

- 0010 - L'utilisateur doit pouvoir prendre une vidéo
- 0020 - L'utilisateur doit pouvoir sélectionner une vidéo depuis sa galerie de médias.
- 0030 - L'utilisateur doit pouvoir sauvegarder l'image finale obtenu
- 0040 - L'utilisateur doit pouvoir partager l'image finale obtenue
- 0050 - L'application doit pouvoir enregistrer les anamorphoses précédemment créées.
- 0060 - L'application doit permettre à l'utilisateur de choisir un type d'anamorphose :
  - Gestion du rideau gauche, droite, haut, bas
  - Gestion du rideau diagonal
  - Rideau qui suit une courbe saisie par l'utilisateur
  - Spirale...
- 0070 - L'application doit pouvoir créer une anamorphose en temps réel sur une vidéo.
- 0080 - L'application doit pouvoir créer une anamorphose en format image.
- 0085 - L'application doit pouvoir afficher l'anamorphose
- 0090 - L'utilisateur peut adapter les paramètres de traitement
- 0100 - L'application adapte l'anamorphose en fonction du nombre d'images et la durée de la vidéo

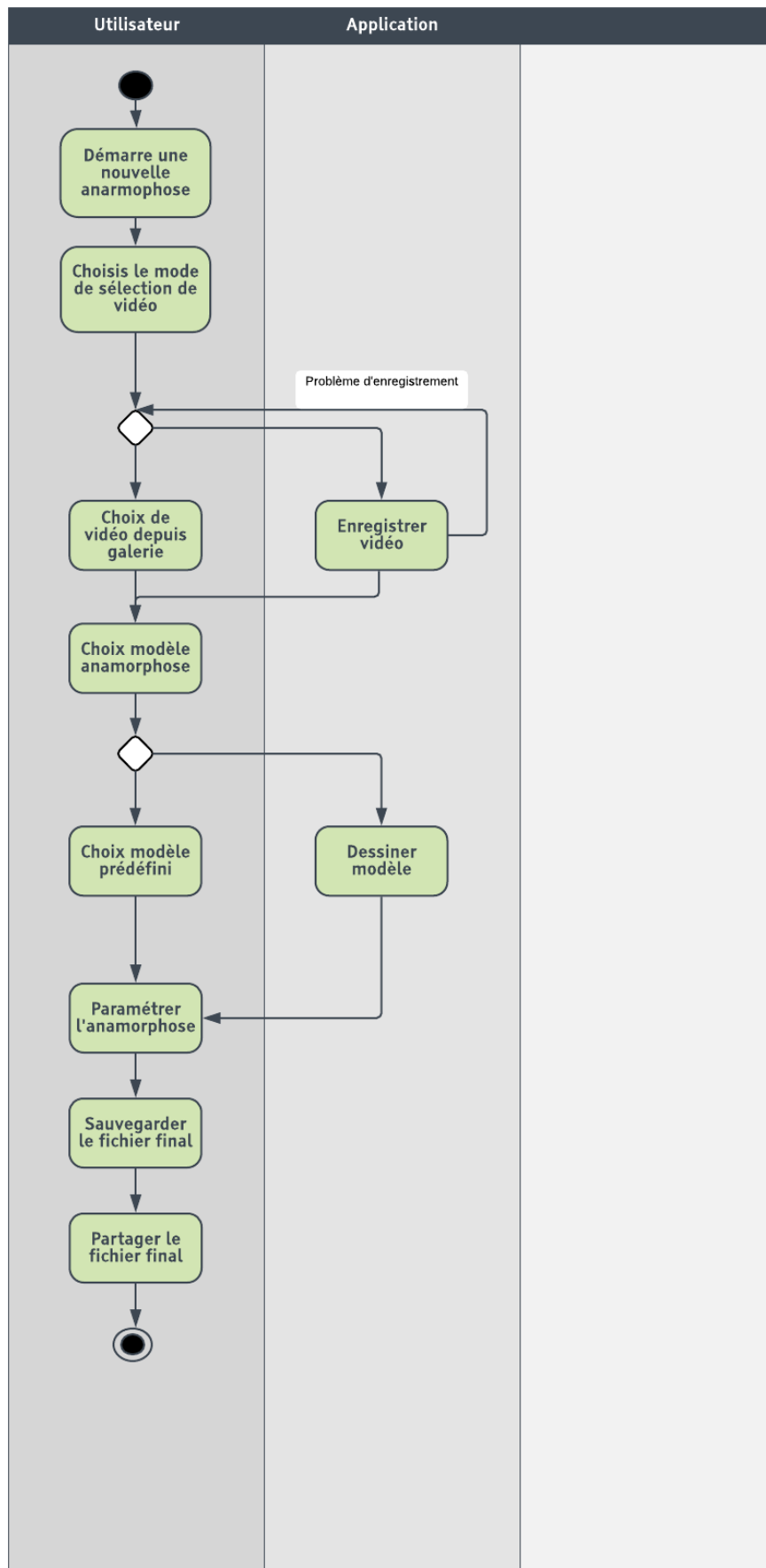
Ces dernières nous ont donc permis de dégager les deux diagrammes principaux correspondants à notre future application.

## 2.2 Diagrammes UML

### 2.2.1 Cas d'utilisation



## 2.2.2 Diagramme d'activité



## 2.3 Problèmes et solutions proposées

La principale problématique est de faire en sorte que notre application soit fonctionnelle pour tous les types de vidéos. C'est à dire qu'il faut que cette dernière soit capable de prendre en compte les caractéristiques variables des différents modèles de téléphones. En effet, d'un téléphone à l'autre, la définition, le nombre d'image par seconde ou encore la résolution peuvent varier et les paramètres de l'anamorphose sont liés à ces caractéristiques. La durée de la vidéo à traiter est également un élément important à prendre en compte, puisque c'est cette dernière qui permet de calculer le nombre d'image totale à traiter.

Mise en situation :

Nous souhaitons obtenir une image anamorphosée de 2Mpx de définition 1920x1080, à partir d'une vidéo de même définition et de 30 images par secondes.

Considérons une anamorphose horizontale, de la gauche vers la droite :

Pour obtenir l'image souhaitée, si l'on décide de prélever une colonne de pixels par image, il nous faudrait 1920 images. Si l'on considère que notre vidéo possède les caractéristiques décrites précédemment (à savoir 1920x1080/30fps), le temps optimal  $t$  pour la vidéo doit vérifier la formule suivante :

$$t = (L / hc) / f$$

où :

$L$  = Largeur de la vidéo,

$hc$  = Nombre de colonne de pixel à prélever par image,

$f$  = nombre d'image par seconde de la vidéo.

Dans notre exemple on a donc  $t = (1920/1)/30 = 64$  secondes.

La formule présentée ci-dessus peut également être généralisée dans le cas d'une anamorphose verticale :

$$t = (H / hl) / f$$

où cette fois on a :

$H$  = Hauteur de la vidéo,

$hl$  = Nombre de ligne de pixel à prélever par image.

Cependant, l'application d'anamorphose que nous devons réaliser doit être capable de traiter toutes les vidéos et pas uniquement celle vérifiant les conditions précédentes.

Si la vidéo à traiter a une durée inférieure au temps calculé, la solution la plus simple consiste à prélever plusieurs colonne/ligne de pixels par image. L'inconvénient est que l'image finale obtenue risque d'être crénelée (aliasing). Une autre solution serait de réduire la définition de l'image finale, mais encore une fois la qualité sera amoindrie et un problème se pose concernant le nombre et le choix des pixels à conserver pour l'image finale.

Si la vidéo à traiter a une durée supérieure au temps optimal calculé, l'idée est de sauter certaines images de la vidéo de manière à garder uniquement le nombre d'image nécessaire à la réalisation de l'image anamorphosée. Pour cela, nous avons réalisé une première version d'algorithme :

Algorithme de sélection de frame:

Entrées:

fps, temps, nbImagesOpti: entiers

Début:

nbFrame <- fps \* temps

pas <- nbFrame / nbImagesOpti

vEntiere <- valeurEntiere(pas)

vReste <- (nbImagesOpti \* valeurDecimale(pas)) / nbFrame

vEcompteur <- vEntiere

vRcompteur <- vReste

Pour i allant de 0 à nbFrame faire:

Si vRcompteur >= 1 alors:

vEcompteur <- vEcompteur + valeurEntiere(vRcompteur)

vRcompteur <- valeurDecimale(vRcompteur)

fin si

Si vEcompteur = 1 alors:

traiterFrame(i)

vEcompteur <- vEntiere

Sinon:

vEcompteur <- vEcompteur - 1

fin si

vRcompteur <- vRcompteur + vReste

fin pour

Fin.

Cet algorithme consiste à calculer un pas en fonction du nombre d'image de la vidéo fournie et du nombre d'image nécessaire à la réalisation de l'image anamorphosée. Ce pas va nous permettre de déterminer quelles sont les frames de la vidéo que nous devons utiliser et quelles sont celles que nous devons passer.

Prenons deux exemples :

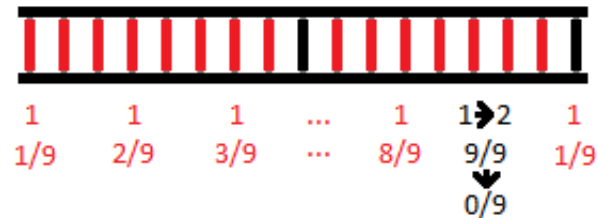
Pour les entrées suivantes :

fps = 30, temps = 72, nbImagesOpti = 1920.

Le pas calculé vaut donc :  $(fps * temps) / nbImagesOpti = (30 * 72) / 1920 = 1.125$ .



Cela signifie donc que une image sur neuf sera sautée car la partie entière vaut 1 (càd qu'on prend toutes les images) et la partie décimale vaut 0.125 on calcule donc un reste de 0.1111 ... . Lorsque ce reste devient supérieur ou égal à 1, cela signifie que l'on doit sauter une image. Voici un schéma du déroulement de l'algorithme :



Les barres rouges correspondent aux images que l'on utilise pour l'image anamorphosée, les bandes noires correspondent à celles que l'on saute.

Pour les entrées suivantes :

$\text{fps} = 30$ ,  $\text{temps} = 142$ ,  $\text{nbImagesOpti} = 1920$ .

On calcule un pas de  $(30 * 142) / 1920 = 2.21875$ .

Dans ce cas là, la partie entière du pas vaut 2, cela signifie donc que l'on utilisera une image sur deux de la vidéo. De plus, la partie décimale vaut 0.21875 on calcule donc un reste de 0.09859154929. Voici le schéma de représentation de l'algorithme pour les valeurs précédentes :

