BEN HAMOUDA Amel
HARDY Elise
Groupe 3

UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE

Rapport du Projet d'informatique

Sujet : Morphing

Table des matières

I. Objectif du Projet

L'objectif de ce projet est de développer une application graphique pouvant générer des morphings entre deux images de taille 512 x 512.

Le morphing généré est contrôlé par des points de repères placés par l'utilisateur sur les deux images.

Le couple de point indiquant où devra arriver l'image d'arrivée sur l'image de départ.

II. Élaboration du Projet

A. Modules

Pour remplir l'objectif de ce Projet on a donc utilisé 5 modules :

→ ***graphique***, qui effectue l'affichage de chaque détails sur la fenêtre graphique ;

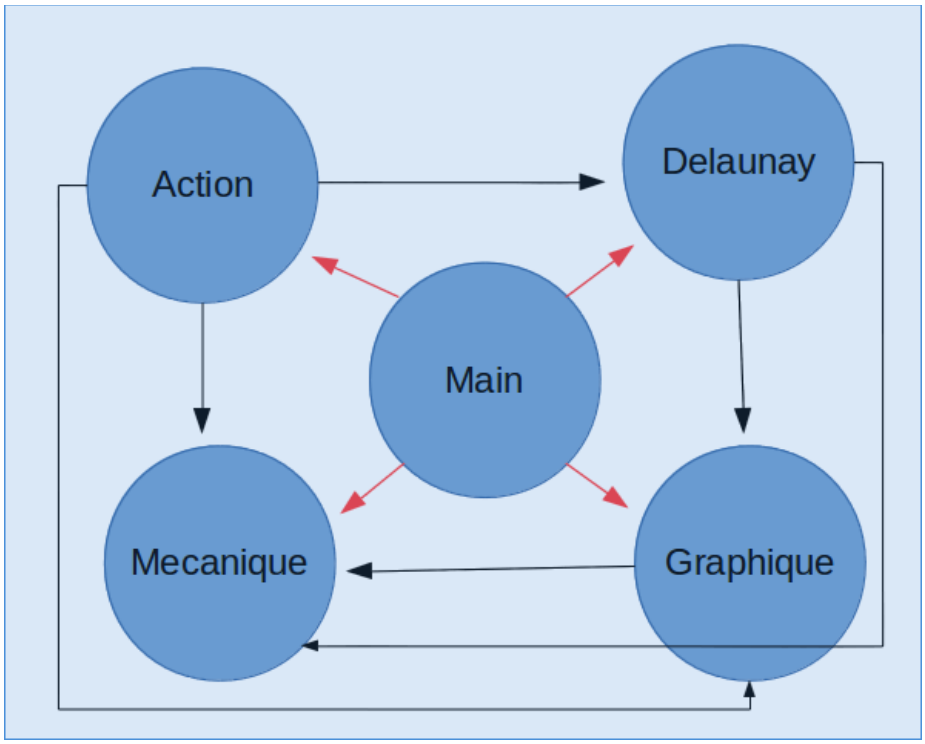
→ ***mecanique***, qui récupère les informations nécessaire en fonction du clic de l'utilisateur ;

→ ***action***, qui effectue l'action nécessaire en fonction de l'information récupérée par le module mécanique ;

→ ***Delaunay***, qui effectue toute la partie mathématique avec les coordonnées barycentrique et le triangle de Delaunay ainsi que la génération des frames;

→ ***main***, qui effectue l'assemblage des 4 modules ci dessus et lance le morphing. Ce module contient la boucle principale du programme.

B. Graphe d'inclusions



C. Choix de programmation

1. Structures

Nous avons utiliser des structures dans le module:

→ ***Delaunay***, avec la matrice et les coordonnées barycentrique

```
typedef struct {
  int n;
  int p;
  long** mat;
} Matrix;
```

&

```
typedef struct {
  float alpha;
  float beta;
  float gamma;
} BarycentricCoordinates;
```

→ ***mecanique***, avec

```
typedef struct {
  int x;
  int y;
} Clic, Point;
```

Un clic qui représente le clic de l'utilisateur

```
typedef struct {
    Clic c1;
    Clic c2;
} Couple;
```

Un couple de clic pour les deux images

```
typedef struct{
  Couple cp1;
  Couple cp2;
  Couple cp3;
} Triangle;
```

Un triangle

```
typedef struct{
    Triangle* t;
    int size;
    int tMax;
} TabTriangle;
```

Un tableau de triangle

```
typedef struct {
  Point p;
  int red;
  int green;
  int blue;
  int alpha;
} ColorPixel;
```

Une couleur pixel pour effectuer le morphing de fin

```
typedef struct {
  int nbFrame;
  MLV_Image *picture1;
  MLV_Image *picture2;
  int active;
  Point **listPoint;
} Board;
```

Un plateau pour la fenêtre graphique

2. Partie mathématiques

Pour la partie mathématiques :

→ **Triangle ordre Anti-horaire :**

Pour déterminer si on prend un triangle ABC dans le bon sens ou non on utilise la formule suivante :

$$(Cx - Ax) * (By - Ay) - (Bx - Ax) * (Cy - Ay)$$

Si ce calcul est supérieur à zéro on ne change rien sinon on inverse les points B et C.

→ **Coordonnées barycentriques :**

Pour calculer les cordonnées barycentriques nous avons utilisé la formule suivante

alpha = aire(PBC)/aire(ABC)

bêta = aire(APC)/aire(ABC)

gamma = aire(ABP)/aire(ABC)

Ce qui revient à calculer des déterminants de matrice 2x2.

→ **Déterminant matrice :**

matrice 2x2 et 3x3 : Pour calculer le déterminant d'une matrice 2x2 et celui d'une matrice 3x3 nous avons utilisé la règle de Sarrus, c'est à dire

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \qquad A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad |A| = a(ei - fh) - b(di - fg) + c(dh - eg)$$

→ **Critère de Delaunay :**

Pour déterminer le critère de Delaunay d'un quadrilatère normalement on utilise une matrice 4x4 mais nous utiliserons ici son équivalent en matrice 3x3

$$= \begin{vmatrix} A_x - D_x & A_y - D_y & (A_x^2 - D_x^2) + (A_y^2 - D_y^2) \\ B_x - D_x & B_y - D_y & (B_x^2 - D_x^2) + (B_y^2 - D_y^2) \\ C_x - D_x & C_y - D_y & (C_x^2 - D_x^2) + (C_y^2 - D_y^2) \end{vmatrix}$$

On calcul le déterminant de cette matrice et ci celui ci est inférieur à zéro alors le quadrilatère contenant les triangles est de Delaunay sinon on doit flippé une des arêtes pour reformer des triangles.

→ **Triangulation déformée :**

Pour calculer la triangulation déformée des triangles la formule données est :

(x, y) → (x', y') devient alors avec le paramètre t de la frame ((1 – t)x + tx', (1 – t)y +ty')

Conversion coordonnées barycentrique en coordonnées cartésiennes :

Pour transformer des coordonnées barycentriques en coordonnées cartésiennes on utilise la formule :

$$x = (\alpha.x_A + \beta.x_B + \gamma.x_C)/(\alpha+\beta+\gamma)$$
$$y = (\alpha.y_A + \beta.y_B + \gamma.y_C)/(\alpha+\beta+\gamma)$$

Lorsque l'on sait où doit aller le point et que l'on à récupéré ses couleurs (rgba image départ et r'b'g'a' image d'arrivée) .

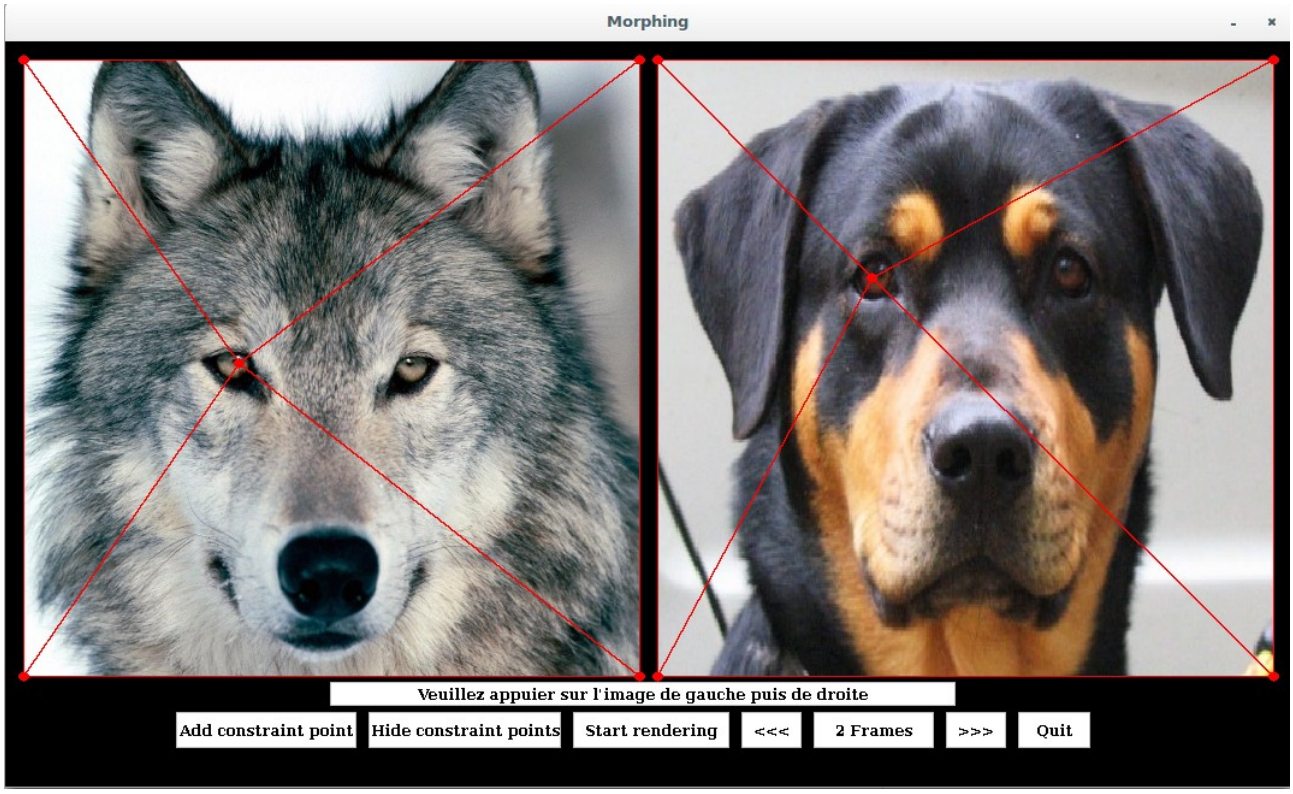
On calcul alors les couleurs interpolées :

((1 – t)r + tr', (1 – t)g + tg', (1 – t)b + tb', (1 – t)a + ta')

3. Graphique

Pour la partie graphique on a suivie le modèle du sujet avec deux images sur un fond noir et les boutons en dessous des images, les lignes et les points sont rouges.

Si on appuie sur l'image de gauche une ligne apparaît sur la fenêtre graphique.



Et pour le résultat final du morphing on a utiliser MLV_get_pixel_on_image avec l'aide de la structure ColorPixel. Cette méthode est utilisée dans la fonction generateFrame dans le module Delaunay.

4. Bugs

Sur certain ordinateur lors du placement de certain points une erreur de segmentation se produit mais sur d'autre ordinateur comme ceux de la faculté tous se passe correctement.
Nous n'avons malheureusement pas réussit à trouver le problème.
La structure Board contient une liste de point qui n'est pas utilisée dans le reste du programme mais quand on essaie de le retirer de la structure. Cela génère une erreur de segmentation.

5. Choix non retenu

Au début on pensait utiliser une structure globale pour que toutes les fonctions puissent connaître la liste de triangle et les autres données. Mais on a rejeté l'idée tous de suite car c'était trop compliqué à gérer.

On a aussi pensé à une liste de point mais la liste de triangle contenant déjà les points, la liste de point ne servait plus à rien.

III. Conclusion

Dans un premier temps la difficulté était sur la compréhension de l'utilisation de git avec toutes ces commandes (add, commit, push, pull, ...) et comment faire pour ne pas avoir de conflit quand on déposer un fichier.
Puis dans un second temps la difficulté était portée sur la compréhension de la partie mathématique avec tous les calculs de conversion de données (coordonnées barycentriques, matrices).
Heureusement le professeur nous a donnée de nombreuses indications ainsi que des formules dans le sujet du projet.
On peut dire que pour ce projet on a du lire plusieurs documentations pour comprendre l'utilisation de git et les calculs à faire pour la partie mathématiques du projet.

IV. Annexe

A. Instructions de compilation

Pour compiler le programme avec les modules il suffit de lancer le *makefile* par la commande **make** dans le terminal.

B. Documentations / Instructions d'utilisation

Pour lancer le programme il suffit de taper ./Morphing suivie des images sur le terminal.
Attention néanmoins car les images se trouvent dans le dossier images.
Il faut donc le préciser dans la ligne de commande.
« ./Morphing images/nomImage1 images/nomImage2

Ensuite une fenêtre s'ouvre avec deux images et des boutons en bas.
L'utilisateur à alors le choix entre ajouter une point, cacher les lignes, lancer le morphing, augmenter/diminuer le nombre de frames ou quitter.

Avant de commencer il faut savoir que l'utilisateur doit d'abord cliqué sur l'image de gauche puis sur l'image de droite pour ajouter les points.
Chaque boutons permet de faire une chose :

→ Add constraint point : après avoir cliqué sur le bouton l'utilisateur peut alors choisir où placer le couple de point.
→ Hide constraint points : permet de caché les lignes et les points sur les images.
→ Start rendering : permet de lancer le rendu qui transforme l'image.
→ <</>> : permet d'augmenter ou de diminuer le nombre de frames pour le rendu.
→ Quit : permet de quitter proprement la fenêtre.

Pour pouvoir ouvrir la documentation Doxygène if suffit d'ouvrir le fichier index.html qui ce trouve dans le dossier doc/html.

De plus si vous téléchargez le projet via le redmine il ne faut pas oublier d'ajouter le répertoire bin, puisqu'on ne peut déposer un répertoire vide sur redmine.