



école supérieure d'art  
du Nord-Pas de Calais/  
Dunkerque-Tourcoing



# Rapport de synthèse

**Générateur interactif de vidéos enchevêtrées**



## Présentation du projet artistique

C'est une installation vidéo interactive inspirée de la phénomene de la pesanteur, elle est fondée sur la théorie de l'anthropocène. Donc la force de la gravité impact la terre, et les mouvements et les choix des humains influencent aussi le monde de nos jours. Dans cette installation, une interface qui présente quelques phrases y compris neuf mots rouges liée avec un kinect, permet aux spectateurs devant cette interface de choisir 3 mots rouges par les mouvements de la main, mais les gens ne les touchaient pas vraiment, ils montraient juste des mouvements dans l'air. Après les choix, cette interface projection ouvre 3 vidéos superposées et aussi les images mouvement en temps réel de cette personne devant. Le but est de créer une sorte de monde virtuel en illusion mélangeant les vidéos qui compose par les différentes directions.

Référence :

Thierry Kuntzel, *The Waves*

Laure Prouvost, *Vois ce bleu profond te fondre.*

# Cahier des charges et des objectifs

L'objectif de ce projet était de construire une interface comprenant des mots liés à des vidéos. L'utilisateur se tiendrait devant un écran affichant l'interface de mot et une caméra reconnaissant les formes. En passant main droite devant la caméra, l'utilisateur pourrait sélectionner 3 mots. 3 vidéos correspondantes à ces 3 mots seraient enchevêtrées puis affichées à l'écran. L'utilisateur, filmé, apparaîtrait ensuite dans les vidéos.

Pour ce projet, nous avons besoin d'ordinateurs munies du logiciel "processing", et d'une caméra de détection de présence "kinect".

Nous avons donc les charges suivantes :

- achat de l'appareil kinect
- lecture et analyse du pré-projet
- configuration des ordinateurs de travaux (installation de processing et de kinect)
- prise en main des technologies
- construction de l'interface tout en tenant compte du design
- test du rendu

# Description des travaux réalisés

analyse du sujet :

Nous nous sommes tout d'abord penchés sur le pré-projet rédigé par Hong Qu. En analysant ce document, plusieurs chemins semblaient possibles. Nous pouvions par exemple filmer l'utilisateur sur fond vert et ajouter sur le fond vert 3 vidéos qu'il aurait sélectionné. Mais une autre possibilité était plus attirante. Il s'agit de capter les mouvements de l'utilisateur avec un appareil Kinect pour sélectionner des vidéos stockées dans processing. Puis faire apparaître l'utilisateur dans les 3 vidéos achevées.

prise en main des outils à disposition :

Lors de ce projet, nous avons principalement travaillé sur le programme sous "processing" permettant de réaliser une interface fonctionnelle et stylisée. Nous avons d'abord pris connaissance des différentes bibliothèques auprès de Stéphane Cabée, et en nous aidant d'internet.

écriture du texte :

Nous avons commencé la programmation de l'interface par l'écriture du texte. La gestion des coordonnées des textes était laborieuse (de nombreuses coordonnées à définir à la main) car nous n'avons pas trouvé de bibliothèques réalisant des tâches spécifiques à nos besoins. Nous avons donc commencé par écrire 3 mots liés à des vidéos pour essayer notre programme sans risquer de changer toutes les coordonnées à cause d'un problème de résolution. Et en fin de projet, nous ajouterions les coordonnées manquantes.

Nous nous sommes ensuite questionné sur les couleurs du texte, et avons choisi le bleu pour le texte non lié à une vidéo, le rouge pour un mot lié à une vidéo et le jaune quand l'utilisateur passerait sur un mot pour le sélectionner.

On ne voit pas la lune pendant la journée, mais la marée du bord de mer est toujours visible. L'attraction de la lune change beaucoup de choses sur notre planète

Gestion des vidéos :

Ensuite, nous sommes passés à la gestion des vidéos.

Nous avons donc programmé quelques conditions.

Par exemple, toutes les vidéos sont mises en variables dans processing et le fait de passer sur un mot lié sélectionne une vidéo.

Par choix artistique, nous voulions qu' à la fin des vidéos, on retourne automatiquement sur le texte de sélection des vidéos. Nous avons donc ajouté la gestion des frames d'une vidéo pour savoir quand retourner sur le texte.

kinect :

Enfin nous avons pu ajouter les fonctionnalités kinect pour filmer les personnes et reconnaître la sélection d'un mot (en passant la main droite dessus).

Pour cela nous avons installé les librairies de kinect pour processing, et avons branché l'appareil kinect à l'ordinateur pour essayer notre programme.



# **Bilan des problèmes non résolus et liste des tâches à faire d'ici l'accrochage**

Il nous reste plusieurs problèmes encore non résolus.

Tout d'abord les vidéos prennent beaucoup de ressources et certains ordis ralentissent la lecture. On pourrait réfléchir à comment optimiser la gestion des vidéos.

D'un point de vue artistique, on peut décorer le texte de sélection des vidéos (y ajouter une image de fond par exemple).

Et enfin, il reste encore à rajouter les vidéos restantes au projet.

## **Conclusion**

Ce projet donne un aperçu de ce qu'il est possible de faire avec les technologies "processing" et "kinect". Nous avons pu travailler ensemble pour lier les fonctionnalités aux rendus visuels. Et plus généralement, ce projet permet de prendre les connaissances de deux milieux différents et d'en faire un résultat convaincant.

# Le projet artistique

-reflet

Après le travail de cette semaine, nous avons réalisé une partie de mon projet, pendant ce processus nous avons rencontré beaucoup de questions mais finalement résolu, maintenant je peux tester les effets de ces vidéo superposition.

Dans le premier vue, c'est des phrases et mots rouges présents sur la vidéo projection, quand le spectateur devant vidéo et 'kinect' ,bouge sa main droite dans l'air choisir 3 mots, cette interface change par les 3 vidéo enchevêtrer et aussi la scène réel du spectateur.

Il est créé une vidéo mélange les environnements prépare et en instant. Comme un miroir virtuel suivi beaucoup variable. Chaque différent choix et le camera capture le différent environnement change les vidéos. C'est comme le changement d'humeur des gens, comme la lune dans l'eau et les fleurs dans le miroir, dans un monde complètement illusoire, voir le monde, tout voir et se voir soi-même.





# **Rapport Scientifique :**

## **Générateur interactif de vidéo enchevêtrées**

# Introduction

Ce projet consiste à concevoir une interface permettant d'enchevêtrer des vidéos sélectionnées et de se voir dans ces vidéos.

Pour réaliser cela, nous utilisons les technologies suivantes :

- un appareil "kinect" permet de filmer une personne et de reconnaître ses mouvements
- le logiciel "processing" avec ses bibliothèques pour programmer l'interface principale
- un projecteur pour afficher l'interface et les vidéos.
- un ordinateur muni des technologies précédentes

## Cas d'utilisation

L'interface initiale comprendra un texte dont 9 mots seront liés à des vidéos.

En jouant avec sa main droite devant la caméra kinect, l'utilisateur pourra sélectionner 3 mots parmi ces 9.

Lorsque 3 mots seront sélectionnés, une seconde interface s'affichera.

Les 3 vidéos correspondantes aux mots sélectionnés et le flux de la caméra kinect seront enchevêtrés et affichés sur cette seconde interface.

Donc l'utilisateur et la pièce où il se trouve seront dans les 3 vidéos.

À la fin des vidéos (18 secondes), l'interface initiale s'affiche de nouveau avec le texte.

L'utilisateur peut ainsi sélectionner 3 nouvelles vidéos et répéter l'action indéfiniment.

détails:

- Un cercle représente la main droite de l'utilisateur dans l'interface initiale.
- Si l'utilisateur déplace sa main pour que le cercle soit sur un mot, ce mot change de couleur et rentre dans la sélection des 3 mots.
- un mot sélectionné ne peut pas être sélectionné de nouveau.

# Étude du programme

Processing offre un éditeur de texte permettant de programmer nos interfaces.

On peut décomposer par nous même notre programme en 3 parties :

- Dans la première partie , nous définissons les objets globaux (accessibles partout dans le programme). On y retrouve les import de bibliothèques, les variables globales et les petites méthodes utiles au programme.
- La deuxième partie (méthode setup) initialise les variables principales
- La 3ème partie donne toutes les instructions d’affichage frame par frame de nos interfaces

## Partie 1 :

Nous utilisons principalement certaines variables :

“\_s” contiendra les données utiles sur le corps comme les coordonnées de la main droite,

“movie” est un tableau contenant jusqu’à 3 vidéos sélectionnées

Les variables “lune”, “maree”, etc contiendront toutes les vidéos sélectionnées ou non,

Nous utilisons aussi des variables comme par exemple “vidDuration” pour avoir l’information de fin de vidéo.

La fonction “contains” nous sert à savoir si une vidéo est déjà sélectionnée ou non.

## Partie 2 :

Dans “setup” la variable “kinect” contiendra l’appareil kinect.

On instancie dans cette partie toutes les vidéos.

## Partie 3 :

La partie draw est divisée en deux sous parties divisées par une boucle if (pour les deux interfaces). Une partie sera consacrée à la lecture de vidéo et l’autre à l’affichage du texte et à la sélection de vidéo.

Dans la première sous-partie,

nous utilisons “tint” pour afficher une image d’une vidéo avec un certain pourcentage d’opacité. on choisit d’afficher les vidéos à 130/255 (=51%) d’opacité

et d’afficher l’output de la caméra à 30/255 (=12%) d’opacité.

Le nombre de frame total d'une vidéo est calculé en multipliant la durée d'une vidéo par le nombre de frame par seconde.

Puis on arrête les vidéos avec stop() quand le nombre d'images affichés atteint le nombre de frame total.

Dans la deuxième sous-partie, nous affichons le texte avec la fonction text() de processing.

Nous gérons les couleurs avec la fonction fill()

Et grâce à la variable "rightHandPos" nous vérifions si les coordonnées de la main droite de l'utilisateur sont dans la zone d'un mot. Si elles y sont, nous ajoutons le mot à la sélection.

## Annexe :

```
// partie 1
// import de librairies
import processing.video.*;
import java.util.Arrays;
import kinect4WinSDK.Kinect;
import kinect4WinSDK.SkeletonData;

// kinect
Kinect kinect;
ArrayList <SkeletonData> bodies;
color[] bodiesColor = {color(255, 0, 0), color(0, 255, 0), color(0, 0, 255), color(#FFC400),
color(#8B00FF), color(#FF0D00)};
int kinectLastZ = 50000;
PVector rightHandPos = new PVector();

// les filmes selectionnés
Movie movie[] = new Movie[3];

// les filmes
Movie lune, maree, planete, visibles;

// nombre de vidéo est selectionnée
int n_lire = 0;

boolean onRead = false;

// gestion des frames de la video
float vidDuration;
float vidFramerate;
int myVideoTotalFrame;
int imgProcess;
```

```

boolean contains(Object tab[], Object e){
    boolean res = false;
    for(int i=0;i<tab.length;i++){
        if(e==tab[i]){
            res = true;
        }
    }
    return(res);
}

void movieEvent(Movie m) {
    if(m==movie[0]){
        imgProcess += 1;
    }
    m.read();
}

// gestion des points et coordonnées de la main droite
void drawJoint(SkeletonData _s, int _j1) {
    noStroke();
    int bodiesId = int(map(_s.dwTrackingID, 0, 100, 0, 5));
    //println(_s.dwTrackingID);
    if(bodiesId < bodiesColor.length && bodiesId < 6){
        fill(bodiesColor[bodiesId]);
        if (_s.skeletonPositionTrackingState[_j1] !=
Kinect.NUI_SKELETON_POSITION_NOT_TRACKED){
            pushMatrix();
            translate(_s.skeletonPositions[_j1].x*width, _s.skeletonPositions[_j1].y*height);
            float jointWeight = (kinectLastZ-_s.skeletonPositions[_j1].z)/1000;
            float jointSat = map(jointWeight, 0, 50, 0, 255);
            fill(hue(bodiesColor[bodiesId]), jointSat, brightness(bodiesColor[bodiesId]));
            ellipse(0, 0, jointWeight, jointWeight);
            popMatrix();
            rightHandPos.x = _s.skeletonPositions[_j1].x*width;
            rightHandPos.y = _s.skeletonPositions[_j1].y*height;
        }
    }
}

void appearEvent(SkeletonData _s)
{
    if (_s.trackingState == Kinect.NUI_SKELETON_NOT_TRACKED)
    {
        return;
    }
    synchronized(bodies) {
        bodies.add(_s);
    }
}

void disappearEvent(SkeletonData _s)

```

```

{
  synchronized(bodies) {
    for (int i=bodies.size()-1; i>=0; i--)
    {
      if (_s.dwTrackingID == bodies.get(i).dwTrackingID)
      {
        bodies.remove(i);
      }
    }
  }
}

```

```

void moveEvent(SkeletonData _b, SkeletonData _a)
{
  if (_a.trackingState == Kinect.NUI_SKELETON_NOT_TRACKED)
  {
    return;
  }
  synchronized(bodies) {
    for (int i=bodies.size()-1; i>=0; i--)
    {
      if (_b.dwTrackingID == bodies.get(i).dwTrackingID)
      {
        bodies.get(i).copy(_a);
        break;
      }
    }
  }
}

```

```

// partie 2
void setup() {
  size(1920,1080);

  kinect = new Kinect(this);
  smooth();
  bodies = new ArrayList<SkeletonData>();

  //colorMode(HSB, 255);

  lune = new Movie(this, "lune.mp4");
  lune.noLoop();
  maree = new Movie(this, "maree.mp4");
  maree.noLoop();
  planete = new Movie(this, "planete.mp4");
  planete.noLoop();
  visibles = new Movie(this, "visibles.mp4");
  visibles.noLoop();

  textSize(30);
}

```

```

// partie 3
void draw() {

    noStroke();
    if(n_lire==3){

        if(!onRead){ // il ne faut faire qu'une seule fois loop (trouver une autre manière que
mousePressed)
            background(255);
            movie[0].play();
            movie[1].play();
            movie[2].play();
            onRead = true;
        }

        vidDuration = floor(movie[0].duration());
        vidFramerate = movie[0].frameRate;
        myVideoTotalFrame = floor(vidDuration * vidFramerate);

        tint(255,130);
        image(movie[0], 0, 0, width, height);
        tint(255,130);
        image(movie[1], 0, 0, width, height);
        tint(255,130);
        image(movie[2], 0, 0, width, height);
        tint(255,30);
        image(kinect.GetImage(), 0, 0, width, height);
        // arret video

        if(imgProcess >= myVideoTotalFrame){
            onRead=false;
            movie[0].stop();
            movie[1].stop();
            movie[2].stop();
            movie[0] = null;
            movie[1] = null;
            movie[2] = null;
            n_lire=0;
            background(255);
            imgProcess = 0;

        }

        }else{
        for (int i=0; i<bodies.size(); i++) {
            //drawJoint(_s, Kinect.NUI_SKELETON_POSITION_HAND_LEFT); ////MAIN GAUCHE
            drawJoint(bodies.get(i), Kinect.NUI_SKELETON_POSITION_HAND_RIGHT);////MAIN
DROITE
        }
    }
}

```

```

fill(0,0,255);
text("On ne voit pas", 10, 80);
fill(255,0,0);
if( rightHandPos.x > 230 && rightHandPos.x <330 && rightHandPos.y > 40 &&
rightHandPos.y < 80){
fill(255, 255, 0);
if(!contains(movie, lune) && !(n_lire==3)){
movie[n_lire]=lune;
n_lire+=1;

}
}
text("la lune", 230, 80);
fill(0,0,255);
text("pendant la journée, mais", 341, 80);
fill(255,0,0);
if( rightHandPos.x >710 && rightHandPos.x <840 && rightHandPos.y > 40 && rightHandPos.y
< 80){
fill(255, 255, 0);

if(!contains(movie, maree) && !(n_lire==3)){
movie[n_lire]=maree;
n_lire+=1;

}
}
text("la marée", 710, 80);

fill(0,0,255);
text(" du bord de mer est toujours visible.", 841, 80);
text("L'attraction de la lune change beaucoup de choses sur notre ", 10, 120);
fill(255,0,0);
if( rightHandPos.x >900 && rightHandPos.x <1000 && rightHandPos.y > 80 &&
rightHandPos.y < 120){
fill(255, 255, 0);
if(!contains(movie, planete) && !(n_lire==3)){
movie[n_lire]=planete;
n_lire+=1;

}
}
text("planète", 900, 120);

fill(255,0,0);
if( rightHandPos.x >900 && rightHandPos.x <1000 && rightHandPos.y > 160 &&
rightHandPos.y < 200){
fill(255, 255, 0);
if(!contains(movie, visibles) && !(n_lire==3)){
movie[n_lire]=visibles;
n_lire+=1;
}
}
}

```

```
    text("visibles", 900, 160);  
  }  
}
```