



Interaction et visualisation de données médicales

Projet réalisé par

Maël Gadbois
Damien Guironnet
Clément Ménard
Yoann Neveu-Dérotrie

Projet encadré par

Jean-Baptiste Fasquel

Table des matières

1. Présentation du projet
2. Gestion du projet
3. Schémas des librairies utilisées
4. Visualisation des données médicales
5. Partie LEAP
6. Partie KINECT
7. Différences entre les deux services
8. Améliorations possibles
9. Apports personnels

1. Présentation du projet

Objectifs :

- Environnement médical
- Faciliter travail des médecins
- Manipuler données médicales

Périphériques :

- Kinect
- Leap Motion

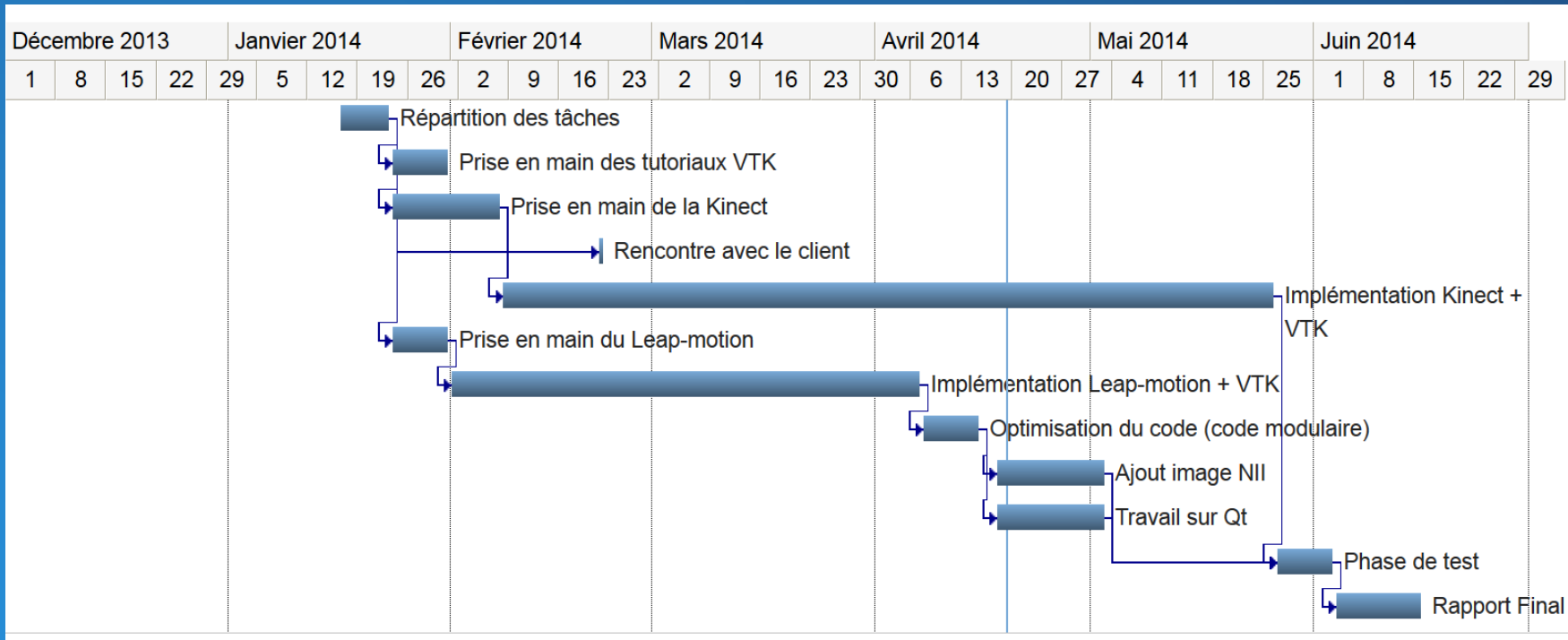
Client :

- Partenariat avec Setúbal

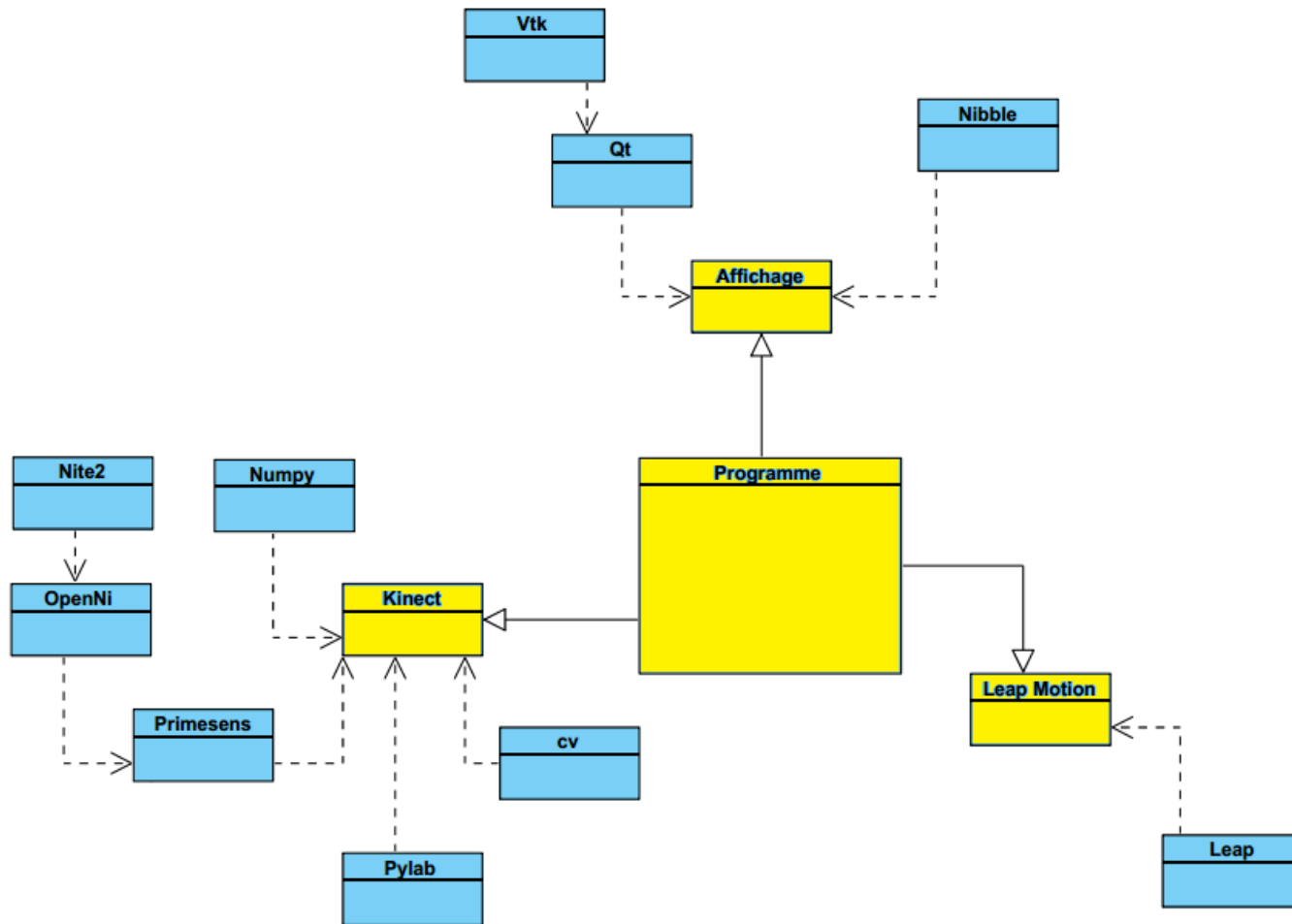


2. Gestion du projet

- Prise en main des tutoriaux
- Division en deux groupes



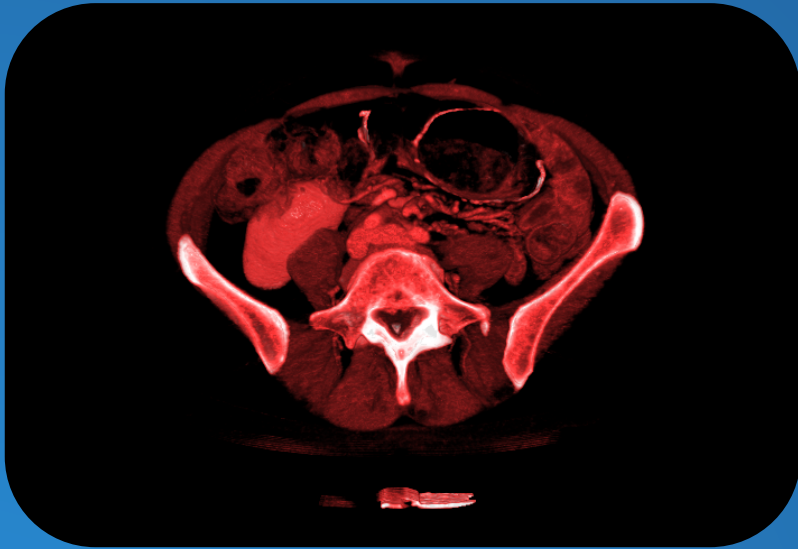
3. Schémas des bibliothèques utilisées



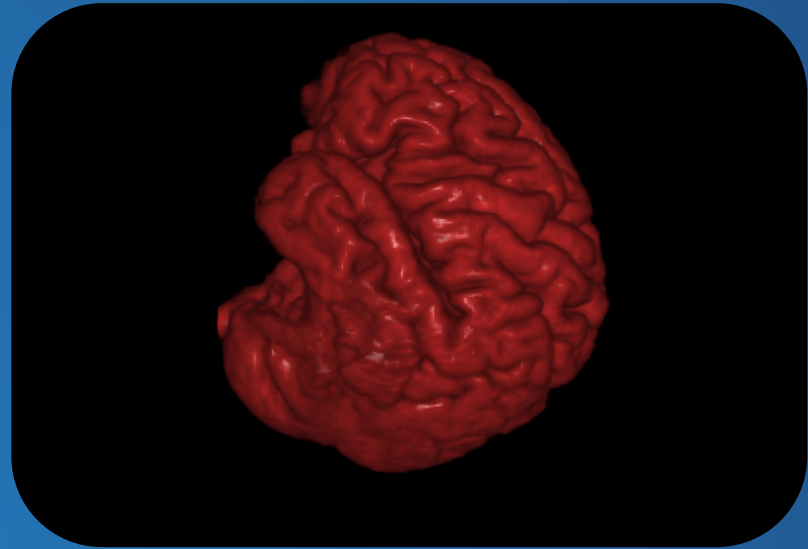
4. Visualisation des données médicales

Deux formats :

Dicom



Nii



Deux manières de visualiser les données :

VTK (Visualization ToolKit)

Une bibliothèque libre pour la visualisation de donnée et le traitement d'image

Dicom (Digital Imaging and Communications in medicine)

-Format dicom géré par vtk

```
dicomReader = vtk.  
vtkDICOMImageReader()  
dicomReader.SetDirectoryName('data')  
dicomReader.Update()  
vtkImage = dicomReader.GetOutput()
```

Nii (NIFTI-1 Data Format")

-Format non géré par vtk
-Librairie Nibabel

```
Image = nibabel.load('im1.nii')  
data = Image.get_data()  
vtkImage = vtkImageImport()  
data_string = data.tostring()  
vtkImage.CopyImportVoidPointer  
(data_string, len(data_string))
```

Fonction de rendu volumique

5. Leap Motion

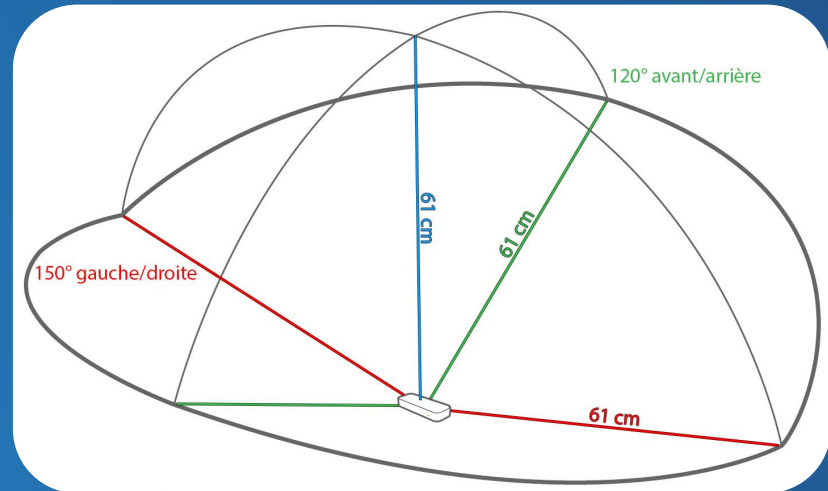
Fonctionnement :

- 2 Caméras infrarouge
- 3 LED infrarouge

Champ de captation : demi-sphère diamètre > 1m

Détection des mains en trois dimensions:

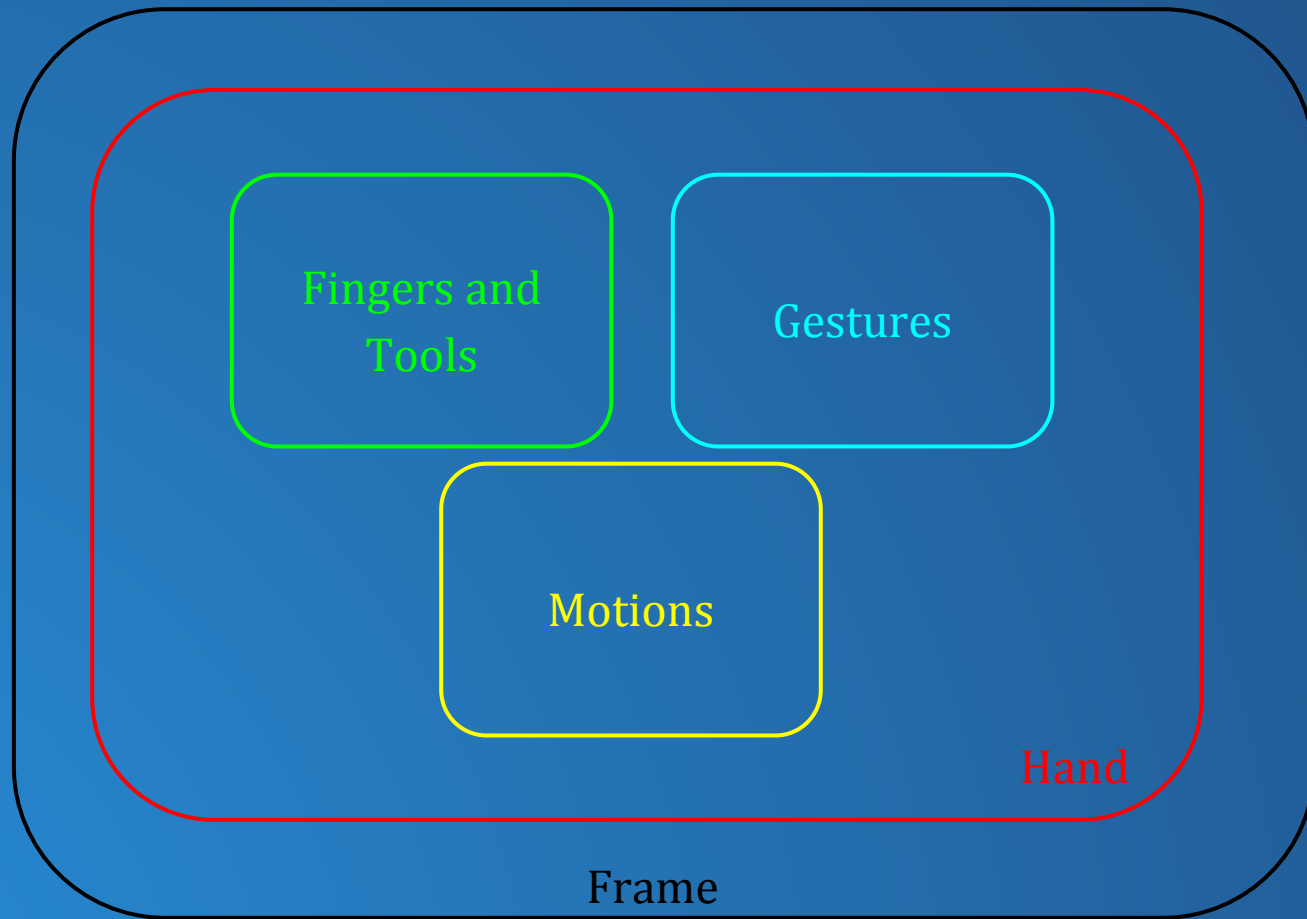
- Nombre de doigts
- Inclinaison
- Courbure
- Orientation
- Gestes simples



Aire de détection du Leap Motion Controller



Différentes classe de l'API Leap Motion :



Déroulement développement

- Utilisation Classes Leap fournies
 - ◆ Informations (Position, Rotation, Gestures...)
 - ◆ Initialisations fournies
- Interaction Objet VTK simple
- Tests Multi-threading : Annulé
- Utilisation des TimerEvents
- Manipulation Image Dicom
- Mise en forme monde 3D
- Interactions personnalisées
- Utilisation doigt comme souris

6. Kinect



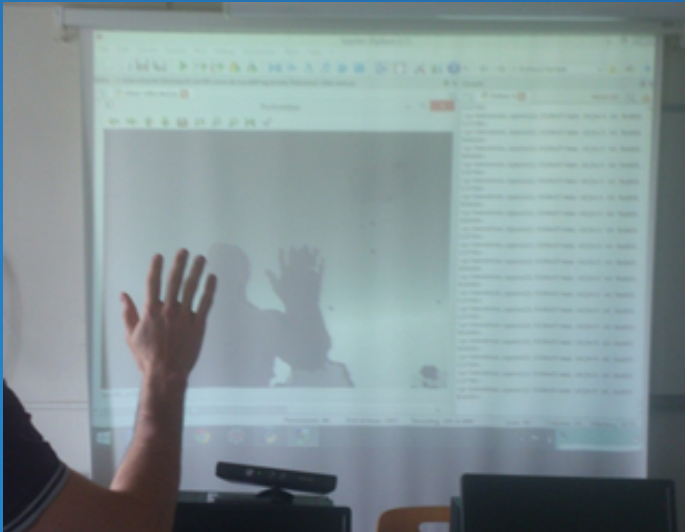
- Caméra RVB
- Détecteur CMOS
- Lumière/caméra infrarouge
- Profondeur perçus 0.8 m 4.0 m

Cahier des charges en fonction des demandes du clients :

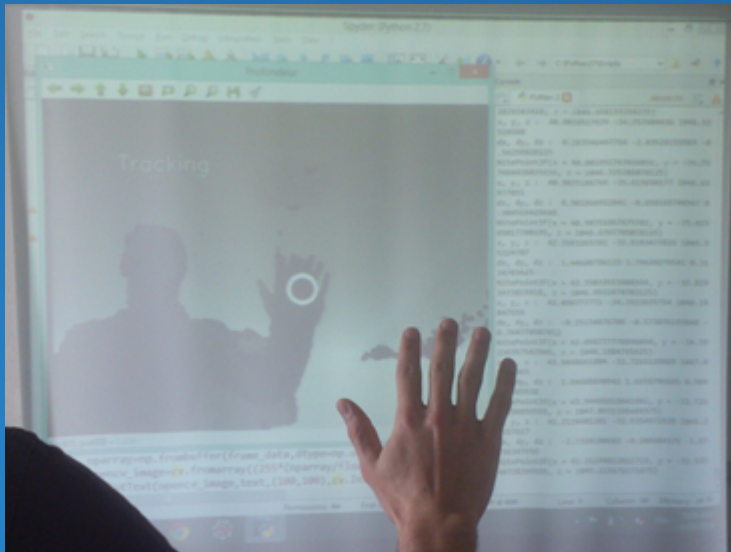
Gestes de l'utilisateur -> mouvements sur l'objet ou l'image

Phase 1 : Détection de la main

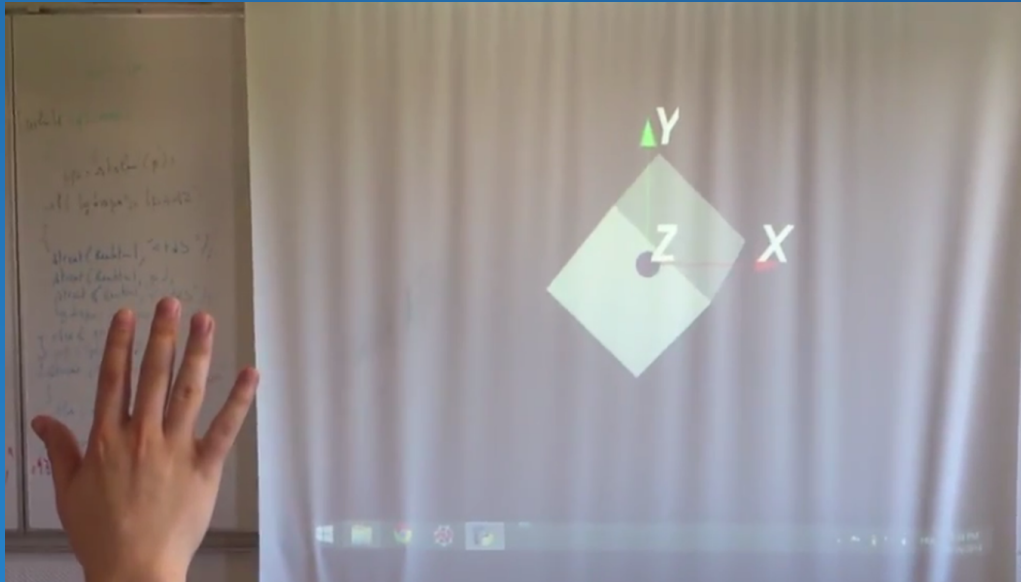
Détecter une main



Phase 2 : Tracking d'une main



Phase 3 : Interaction avec un objet

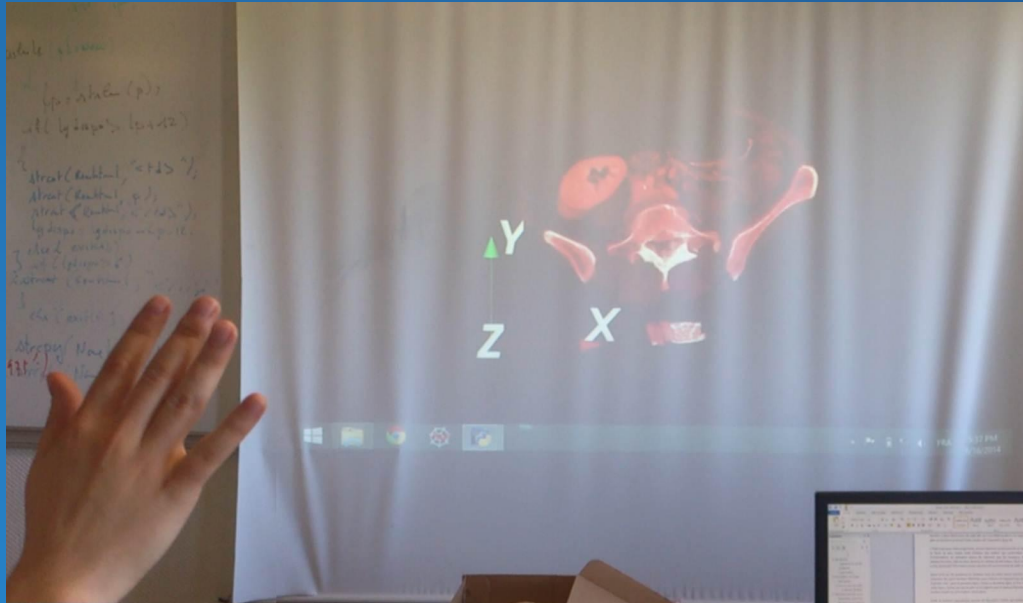


Démos vidéos



Gestes de la main	Mouvements correspondant sur le cube
Rotation autour de l'axe Y	Translation Droite / Gauche
Rotation autour de l'axe X	Translation Haut / Bas
Zoom / Dézoom	Translation Avant / Arrière

Phase 4 : Interaction avec une image médicale



Démos vidéos



Gestes de la main	Mouvements correspondant sur le cube
Rotation autour de l'axe Y	Translation Droite / Gauche
Rotation autour de l'axe X	Translation Haut / Bas
Zoom / Dézoom	Translation Avant / Arrière

Phase 5 : Tracking de deux mains



Démos vidéos



Problèmes rencontrés

- Installation
- Documentation

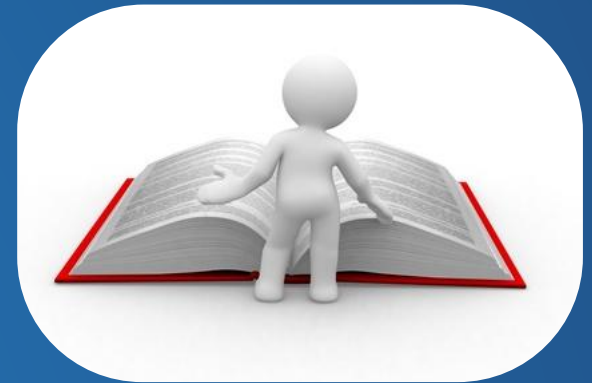


7. Différences entre les deux services

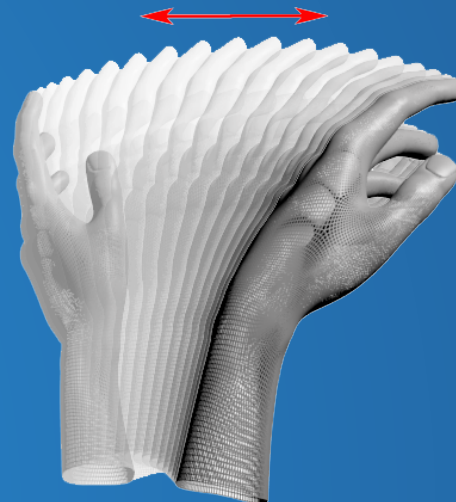
→ Installation des périphériques



→ Documentation de la programmation



→ Détection de la main



8. Améliorations possibles



- Point de vue du Leap Motion
- Point de vue de la Kinect
- QT

Conclusion

