

06 février 2017

CHARRON – CHEVRIER - CREVAN

Projet developpÉ dans le cadre de l’ESEO

SUPPORT 2016-2017

2016/2017

PROJET DE FIN D’ÉTUDE

TRADUCTION DE LA LANGUE

DES SIGNES

SOMMAIRE

[ORGANISATION 3](#_Toc474091106)

[ Outil Microsoft Visual Studio 3](#_Toc474091107)

[Général 3](#_Toc474091108)

[Raccourci 3](#_Toc474091109)

[Partie MainWindow.xaml 4](#_Toc474091110)

[MainWindow.xaml.cs 4](#_Toc474091111)

[ Schéma de conception 5](#_Toc474091112)

[Dossier PROCESS 6](#_Toc474091113)

[Dossier TREATMENT 7](#_Toc474091114)

[METHODES 8](#_Toc474091115)

[BIBLIOGRAPHIE 9](#_Toc474091116)

[ Format de la bibliographie 9](#_Toc474091117)

[ Général 9](#_Toc474091118)

[ Reconnaissance du contour, des extrémités d’une main 10](#_Toc474091119)

[ Identification des doigts 11](#_Toc474091120)

# ORGANISATION

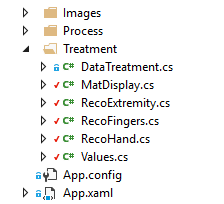
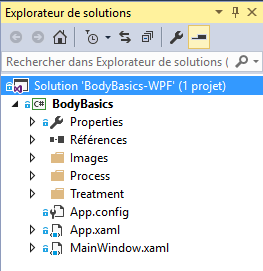
## Outil Microsoft Visual Studio

### Général

Nous allons vous expliquer rapidement comment le code a été organisé pour ce projet au cours de l’année 2016-2017. Il reprend globalement l’organisation de la version précédente de ce projet réalisé en stage technique 2016, effectivement nous avons voulu garder cette idée de conception du code.

Les parties importantes sont : **Treatment**, **Process**, **MainWindows.xaml** et **MainWindows.xaml.cs**.

Le dossier Images ne contient que les images utilisées lors de la démonstration. Chaque dossier comporte plusieurs classes dont nous allons donner une explication.

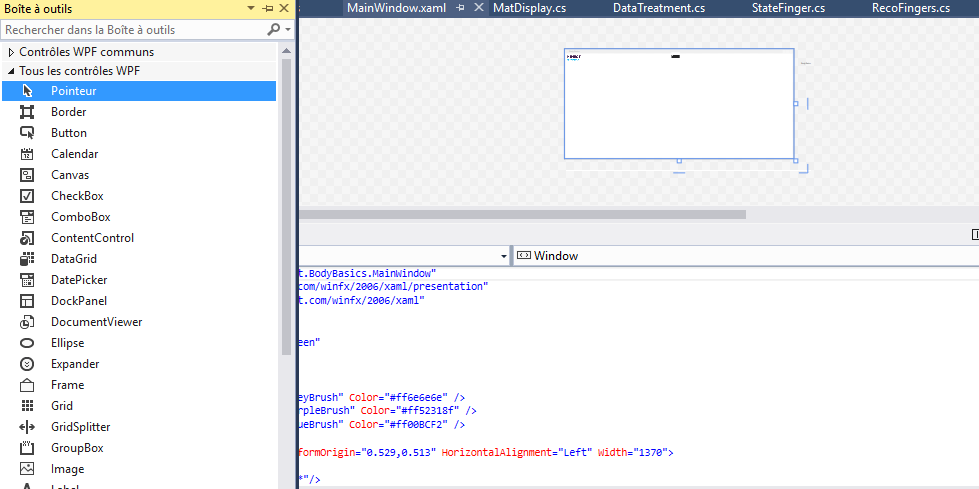


### Raccourci

|  |  |
| --- | --- |
| Commenter du code | Ctrl + k + c |
| Décommenter du code | Ctrl + k + u |

### Partie MainWindow.xaml

Nous allons pouvoir à partir d’ici gérer tout l’affichage de notre démonstration. Ajouter des fenêtres, des boutons, etc. Cette partie est composée de cette manière :



1

2

3

1 : Partie qui concerne l’insertion d’outils sur la partie 2, cette manipulation va se répercuter automatiquement sur la partie 3.

2 : Partie où nous pouvons concevoir notre fenêtre de démonstration. Actuellement nous avons 4 fenêtres miniature et une plus grand qui est notre générale.

3 : Code qui correspond à la structure de l’ensemble. Elle permet une meilleure précision pour les paramétrages des fenêtres mais surtout d’entrer les bonnes références au sein de chacune d’elle.

### MainWindow.xaml.cs

Tout ce qui concerne le bon fonctionnement du programme. Nous allons y retrouver tous les paramètres d’affichage ; des méthodes de constructions de points, de lignes, etc.

* Cette classe est très chargée, peut-être faudrait-il mieux créer une classe « Draw » pour y mettre toute les méthodes concernant l’affichage. Celles-ci sont à la fin du code.

## Schéma de conception

Légende :

À faire

En place

CORPS

MAIN

VISAGE

Point(s) du corps

Point(s) de la main

Structure main

BRA(S)

DOIGT(S)(S)

Structure doigt

Point(s) d’un doigt

## Diagramme de classes

Generated by PlantUML

### Package PROCESS

#### Finger.cs

public struct StructFinger

{

public int indexListPts { get; set; }

public Point extremity { get; set; }

public Point Jointure1 { get; set; }

public Point Jointure2 { get; set; }

public Point otherExtremity { get; set; }

public Point3D extremity3D { get; set; }

public Point3D Jointure13D { get; set; }

public Point3D Jointure23D { get; set; }

public Point3D otherExtremity3D { get; set; }

public Vector3D vOtherToJoint2 { get; set; }

public Vector3D vJoint2ToJoin1 { get; set; }

public Vector3D vJoin1Toextremity { get; set; }

public String name { get; set; }

}

Un doigt est construit avec une structure « StructFinger ».

Les points 3D surlignés en jaune ne sont pas encore utilisés, nous étions en cours de modification du code.

* L’idée est de supprimer les points en deux dimensions si possible pour les remplacer par les point en 3 dimensions.

/! \ Beaucoup de méthodes utilisent les pixels (type Point).

#### Hand.cs

public Hand()

{

listPts = new List<Point>();

listStruct = new List<StructHand>();

fingerHand = new List<Finger.StructFinger>();

center = new Point();

tip = new Point();

thumb = new Point();

wrist = new Point();

/\*Non utilisé pour le moment\*/

center3D = new Point3D();

tip3D = new Point3D();

thumb3D = new Point3D();

wrist3D = new Point3D();

}

**ListPts**

La liste des points du contour d’une main.

**listStruct**

La liste de la structure de la main. Elle va permettre de rassembler tous les points représentant une extrémité ou un creux de la main. La structure est composée d’un point indiquant la position (x,y) et d’un état (Extremity ou Hollow).

**fingerHand**

La liste des doigts de la main. Utilisation de la structure n’est pas forcement la bonne solution.

* L’utilisation d’une liste de Finger pourrait être envisager /! \ Gestion de la structure.

**Center, tip, thumb, wrist**

Les points de la main repérés automatiquement par la kinect version 2. Utilisation pour l’identification des noms des doigts notamment.

#### OurBody.cs

Cette classe n’est pas assez utilisée pour le moment.

* Lier cette classe aux autres. Un corps est composé de deux mains, de certains points du corps et d’un visage. Continuité de la reconnaissance.

#### StateFinger.cs & StateHand.cs

Différents états de la main, des doigts. Nous y retrouvons les noms des doigts par exemple.

### Dossier TREATMENT

#### MatDisplay.cs

Nous y retrouvons la reconnaissance de la main gauche et droite qui va servir pour la suite. Chaque pixel de la Kinect se voit attribuer 1 (G), 2(D) ou 0 (Rien) selon les conditions établies.

Les conditions ont été basées sur la profondeur du centre des mains, à plus ou moins une valeur nous retenons le pixel courant.

#### RecoExtremity.cs

Nous construisons tout ce qui concerne le contour de la main, ses extrémités. Actuellement, la méthode « *searchExtremity()* » récupère bien toutes les informations nécessaire à la construction de la main.

Une deuxième méthode est en cours de développement, l’idée est d’utilisé celle-ci en cas de main fermée

* Nous parcourons d’abord la première, si les résultats sont nuls alors nous parcourons la deuxième. Peut-être serait-il judicieux de regarder du coup des fonctions Kinect pour repérer lorsque la main est ouverte ou fermée (meilleure condition).

#### RecoFinger.cs

#### RecoHand.cs

#### Values.cs

# METHODES

# BIBLIOGRAPHIE

## Format de la bibliographie

Format établi pour cette bibliographie.

|  |  |
| --- | --- |
| **Titre** | |
| AUTEUR(S) | DATE DE PUBLICATION |
| Nom du document | |
| Lien vers le site web | |
| « Idée intéressante du document » | |

## Général

|  |  |
| --- | --- |
| **3D Skeleton extraction using one Kinect camera.** | |
| C.GLYNOS | 2012 |
| Skeleton\_GLYNOS\_2012.pdf | |
| <https://nccastaff.bournemouth.ac.uk/jmacey/MastersProjects/MSc12/Glynos/docs/MasterThesis_CG.pdf> | |
| TRACKING + DESCRIPTION + MODELISATION | |

|  |  |
| --- | --- |
| **Détection d’activité à l’aide de Kinect 2.0** | |
| M.TAZLARI | 2014 |
| / | |
| <http://doc.rero.ch/record/235890/files/TB_Tazlari_Martin.pdf> | |
| TRACKING + GESTURE | |

|  |  |
| --- | --- |
| **Détection et interprétation des Gestes de la Main** | |
| N.MOLLET, R.CHELLALI | 2005 |
| Detection\_Interpretation\_MOLLET\_2005.pdf | |
| <http://www.setit.rnu.tn/last_edition/setit2005/image-video/18.pdf> | |
| DETECTION + CONTOUR + EXTREMITE + MODELISATION | |

## Reconnaissance du contour, des extrémités d’une main

|  |  |
| --- | --- |
| **Fingers and gesture recognition with kinect v2 sensor** | |
| A.LEKOVA, D.RYAN, R.DAVIDRAJUH | October 2016 |
| Extremites\_Bissectrice\_LEKOVA-RYAN-DAVIDRAJUH\_ 2016.pdf | |
| <http://www.iser.bas.bg/staff/lekova/fingers.pdf> | |
| EXTREMITES + BISSECTRICE | |

|  |  |
| --- | --- |
| **Finger Application using K-Curvture Method and Kinect sensor in real-time** | |
| B.ZABRI ABU, S.ROSDIYANNA, P.DWI, M.MAHFUZAH, H.NOR RUL | January 2016 |
| / | |
| <https://www.researchgate.net/publication/288827110_Finger_application_using_K-Curvature_method_and_Kinect_sensor_in_real-time> | |
| K-CURVATURE + FINGER EXTREMITY | |

|  |  |
| --- | --- |
| **DEA IARFA – Techniques du traitement d’images. Description de contours et de formes** | |
| F.TUPIN | / |
| Techniques\_Contour\_TUPIN.pdf | |
| <http://perso.telecom-paristech.fr/~bloch/P6Image/FT_Formes.pdf> | |
| TECHNIQUE CONTOUR | |

|  |  |
| --- | --- |
| **Dossier : Le traitement d’images numériques des bords aux contour des objets** | |
| B.DIALLO | 1996 |
| / | |
| <http://obligement.free.fr/articles/traitement_images_3.php> | |
| TECHNIQUE CONTOUR | |

## Identification des doigts

|  |  |
| --- | --- |
| **Finger identification- based gesture and point cloud-based 3D motion gesture for Natural User Interface** | |
| U.LEE | March 2016 |
| Identification3D\_LEE\_2016.pdf | |
| <http://www.iplab.cs.tsukuba.ac.jp/paper/doctor/lee_doctor.pdf> | |
| ANGLE + VECTEUR + CLASSIFICATION DES DOIGTS | |

|  |  |
| --- | --- |
| **Robust Part-Based Hand Gesture - Recognition Using Kinect Sensor** | |
| Z.REN, J.YUAN, J.MENG, Z.ZHANG | August 2013 |
| Identification3D\_LEE\_2016.pdf | |
| <http://web.cs.ucla.edu/~zhou.ren/TMM_RenYuanMengZhang_submission.pdf> | |
| RECOGNITION FINGER + RECOGNITION HAND + METHOD CURVE | |

|  |  |
| --- | --- |
| **Hand tracking, finger identification and chordic manipulation on a multi-touch surface** | |
| W.WESTERMAN | 1999 |
| Finger\_Identification\_WESTERMAN\_1999.pdf | |
| <https://www.eecis.udel.edu/~westerma/main.pdf> | |
| FINGER IDENTIFICATION + SEGMENTATION HAND + TRACKING | |

## Reconnaissance de mouvements et de signes

|  |  |
| --- | --- |
| **Sign Language Recognition** | |
| Y.SONG et Y.YIN | December 2008 |
| SignLanguageRecognition\_MachingLearningTermPaper\_MITdoc.pdf | |
| <http://people.csail.mit.edu/yingyin/resources/doc/projects/6867term-project.pdf> | |
| HIDDEN MARKOV MODELS | |

|  |  |
| --- | --- |
| **Sign Language Recognition with the Kinect Sensor Based on Conditional Random Fields** | |
| MDPI | December 2014 |
| / | |
| <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4327011/> | |
| RECOGNITION + HIDDEN CONDITIONAL RANDOM FIELDS | |

|  |  |
| --- | --- |
| **American Sign Language Alphabet Recognition Using Microsoft Kinect** | |
| D.CAO, L.MING.C et Y.ZHAOZHENG | 2015 |
| AmericanSignLanguageAlphabetRecognitionUsingMicrosoftKinect.pdf | |
| <http://web.mst.edu/~yinz/papers/CVPR2015Hands.pdf> | |
| SEGMENTATION + RECOGNITION | |

|  |  |
| --- | --- |
| **Accord.NET Framework** | |
| C.SOUZA | 2015 |
| / | |
| <http://accord-framework.net/intro.html> | |
| FRAMEWORK + RECOGNITION | |

|  |  |
| --- | --- |
| **Hidden Conditional Random Fields for Gesture Recognition** | |
| D.MORENCY, D.DEMIRDJIAN | 2015 |
| / | |
| <http://people.csail.mit.edu/sybor/cvpr06_wang.pdf>  <http://crsouza.com/2010/03/23/hidden-markov-models-in-c/> | |
| HCRF MODELS + RECOGNITON + HIDDEN MARKOV MODELS | |

|  |  |
| --- | --- |
| **Hidden Markov Models for Sign Language Recognition : a review** | |
| V-N.PASHALOUDI et K-G.MARGARITIS | 2002 |
| / | |
| <http://lpis.csd.auth.gr/setn02/poster_papers/343.pdf> | |
| RECOGNITON + HIDDEN MARKOV MODELS | |