

Documentation technique

[EIP] TransLSF



Résumé

Ce document présente la partie technique de TransLSF. Il décrit les aspects techniques de notre projet et vise des développeurs ou autres utilisateurs qui pourraient joindre le développement de l'application.

Cette documentation est présentée à travers divers schémas explicatifs.

Il comporte également un lien la documentation du code qui est faite en DOXYGEN.

Ce document vise les développeurs souhaitant rejoindre le projet, pour leur permettre de s'adapter facilement aux règles techniques mise en place par l'équipe déjà présente sur le projet.



Description du document

Titre	2015_TD3_FR_TransLSF		
Date	28/09/2014		
Auteur	Henri HARDILLIER		
Responsable	Henri HARDILLIER		
E-Mail	contact@translsf.fr		
Site Web	http://www.translsf.fr		
Mur du projet	http://eip.epitech.eu/projects/view/translsf		
Gestionnaire de version	https://labeip.epitech.eu/svn/2015/translsf		
Sujet	Documentation technique		
Mots-clés	Langue des signes, handicap, traduction instantanée, Kinect, voix, LSF, EIP, Epitech		
Version du document	3.0		



Tableau des révisions

Date	Auteur	Section(s)	Commentaire
28/09/2014	Henri HARDILLIER	Toutes	Reprise de la version 2.0
01/10/2014	Henri HARDILLIER	Toutes	Refonte du document + ajout des UMLS
03/10/2014	Léo MARTIN	Toutes	Résumé + correction des fautes



Sommaire

I. Introduction	1
A. Présentation de l'EIP	1
B. Présentation de TransLSF	2
II. Vue globale du projet	3
A. Cas d'utilisation	3
B. Déploiement	4
III. Vue logique du projet	5
A. Technologies utilisées	5
1. C# / .NET	5
2. SDK Kinect	6
B. Architecture	7
C. Organisation	8
IV. Vue physique du projet	9
A. L'interface graphique utilisateur	9
B. Les controllers	10
C. Le gestionnaire de Kinect	11
D. Les données	12
1. Les signes	12
2. Les gestes	13
E. Le gestionnaire de données	14
V. Environnement de développement	15
A. IDE	15
B. Gestionnaire de version	16
VI Dogumentation du code course	100



I. Introduction

A. Présentation de l'EIP

Dans le cadre du cursus Epitech, les étudiants doivent mener à terme un projet sur une durée de 18 mois : l'EIP (Epitech Innovative Project). Cela leur permet d'acquérir une expérience de gestion intégrale de projet par la pratique.

Le projet doit avoir une vision professionnelle, c'est-à-dire, qu'à la fin du projet celui-ci doit être complètement fonctionnel.





B. Présentation de TransLSF

TransLSF est un logiciel de traduction instantanée de la langue des signes française. Il est destiné à toute la communauté sourde et malentendante de France.

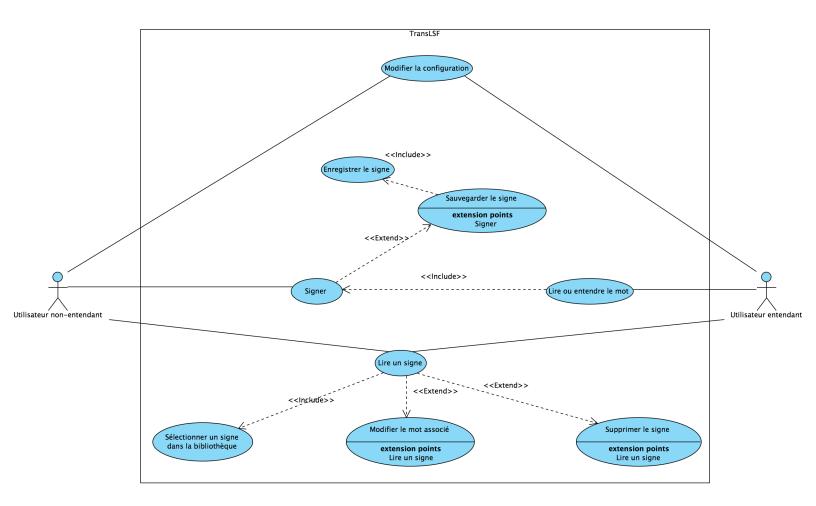
Grâce à une caméra Kinect de Microsoft, il permet de transcrire les signes faits devant la Kinect sous forme écrite ou orale. Ceci permet à l'utilisateur de bénéficier d'une plus grande autonomie dans sa communication avec autrui.



II. Vue globale du projet

A. Cas d'utilisation

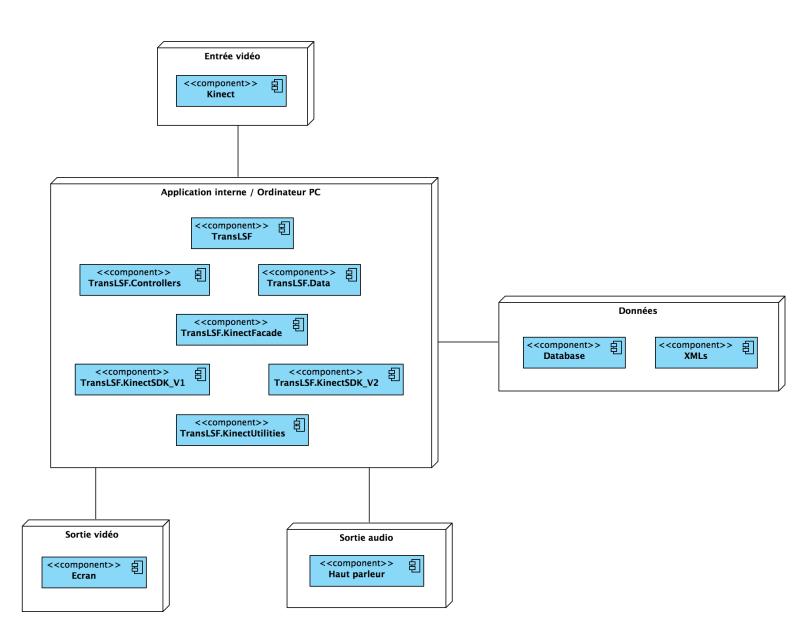
Ci-dessous, voici un diagramme de cas d'utilisation qui représente les interactions fonctionnelles entre les acteurs et le système TransLSF :





B. Déploiement

Ci-dessous, voici un diagramme de déploiement qui représente les liens entre les ressources matérielles et les abstractions logiques que sont les différents composants de TransLSF :





III. Vue logique du projet

A. Technologies utilisées

1. C# / .NET

Nous avons choisi le C#, et plus particulièrement le framework .NET, pour développer notre projet, que ce soit l'API de traduction, ou l'interface utilisateur.

Le C# est un langage inventé par Microsoft. La syntaxe et les concepts de JAVA ont beaucoup inspiré la création de .NET, mais le C# possède un avantage, la possibilité de surcharger des opérateurs. Le .NET, également crée par Microsoft, est un framework C# permettant d'exploiter au maximum le C#. Il nous permet également de gérer une base de données de l'application sans avoir besoin d'internet.





2. SDK Kinect

Le projet étant compatible avec les deux Kinect, nous utilisons deux versions du SDK, la version 1.0 et 2.0, respectivement pour la Kinect 1 et Kinect 2.

Le SDK nous permet d'interpréter les mouvements réalisés par un utilisateur, les reconnaitre, les traduire, et les retranscrire à l'écran. Chaque mouvement dans l'espace enregistré par la Kinect est transmis à notre API de traduction qui renvoie une transcription conforme au signe exécuté.

La Kinect étant également développée par Microsoft, le langage utilisé est le C#.





B. Architecture

TransLSF se divise en plusieurs parties. Chaque partie est représentée par un package de classe qui se défini par un namespace appartenant à la solution TransLSF.

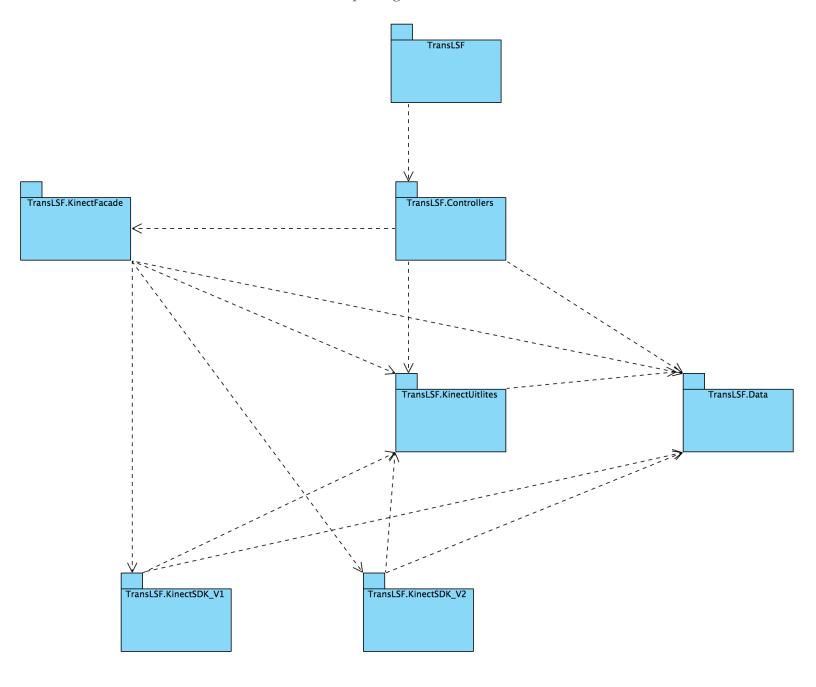
Voici une description des 7 parties de la solution :

- TransLSF : c'est l'interface graphique utilisateur de la solution
- **TransLSF.Controllers** : founit l'ensemble des contrôles fonctionnels (hors GUI) de la solution
- TransLSF.Data : fournit l'ensemble des objets d'accès aux données (DAO) de la solution, ainsi que les objets models de TransLSF
- TransLSF.KinectFacade : fournit une facade des SDKs des Kinects et manage les états de ces périphériques
- TransLSF.KinectSDK_V1: fournit un wrapper sur le SDK 1.8 de la Kinect V1
- TransLSF.KinectSDK_V2 : fournit un wrapper sur la bêta du SKD 2.0 de la Kinect V2
- **TransLSF.KinectUtilities** : fournit un ensemble d'objets models et de types partagé par plusieurs packages différents



C. Organisation

Ci-dessous, voici un diagramme de packages qui représente l'organisation logique de TransLSF et les relations entre packages :





IV. Vue physique du projet

A. L'interface graphique utilisateur

Cette partie du projet, représentée par le namespace TransLSF, représente la GUI du logiciel. Elle fournit tous les contrôles et les actions des éléments de l'interface graphique.

L'architecture du code respecte le pattern MVVM (Model-View-ViewModel) qui est une méthode de conception amenée par Microsoft et qui se base sur une séparation MVC (Model-View-Controller) adaptée aux technologies Microsoft. Elle sépare la vue de la logique et de l'accès aux données en accentuant les principes de binding et d'événements.

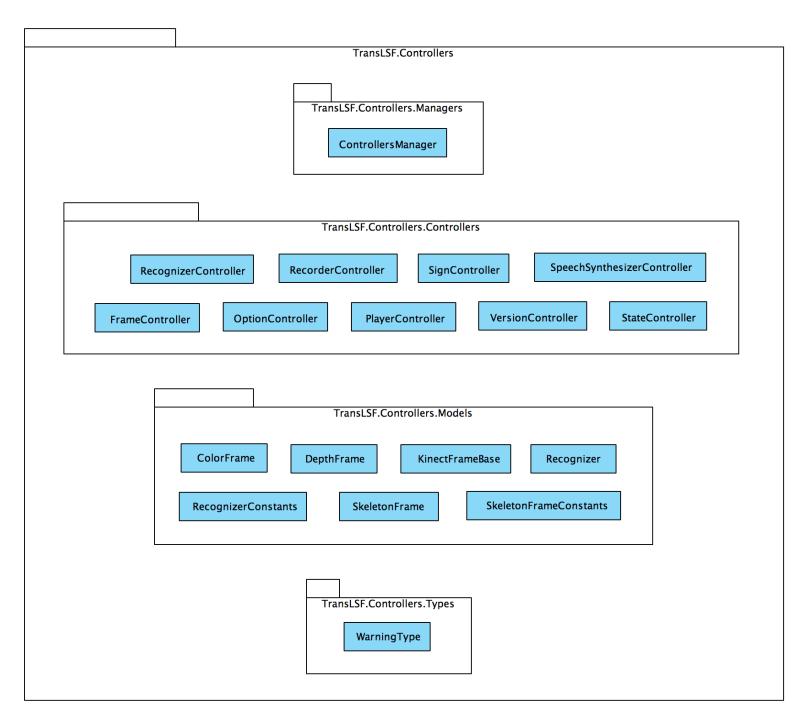
Cette partie est donc composée (entre-autres) des 3 namespaces :

- TransLSF.Models
- TransLSF.Views
- TransLSF.ViewModels



B. Les controllers

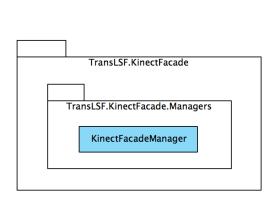
Cette partie fournit l'ensemble des contrôles fonctionnels utilisés par TransLSF. Chacun des controller respecte le SRP (Principe de Responsabilité Unique) et est accessible via le ControllerManager.

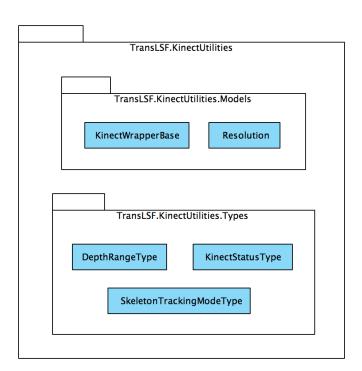


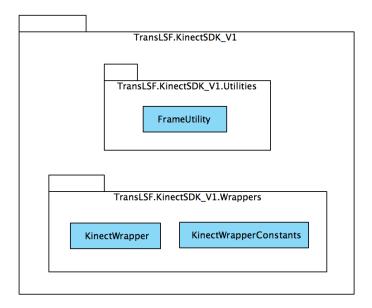


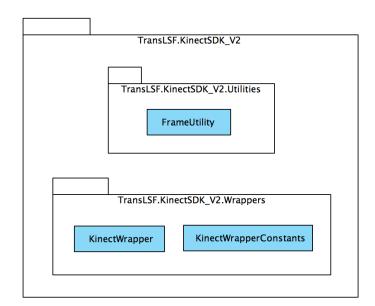
C. Le gestionnaire de Kinect

La gestion des Kinects est fournit par le namespace TransLSF.KinectFacade. Elle s'occupe de fournir une facade des SDKs des Kinects et manage les états de ces périphériques.







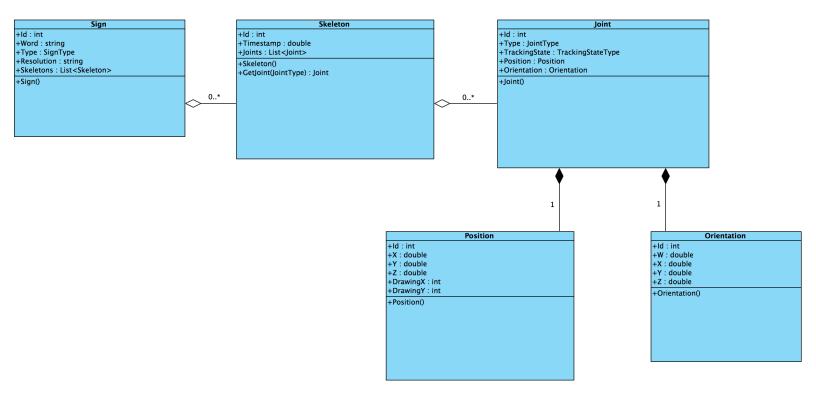




D. Les données

1. Les signes

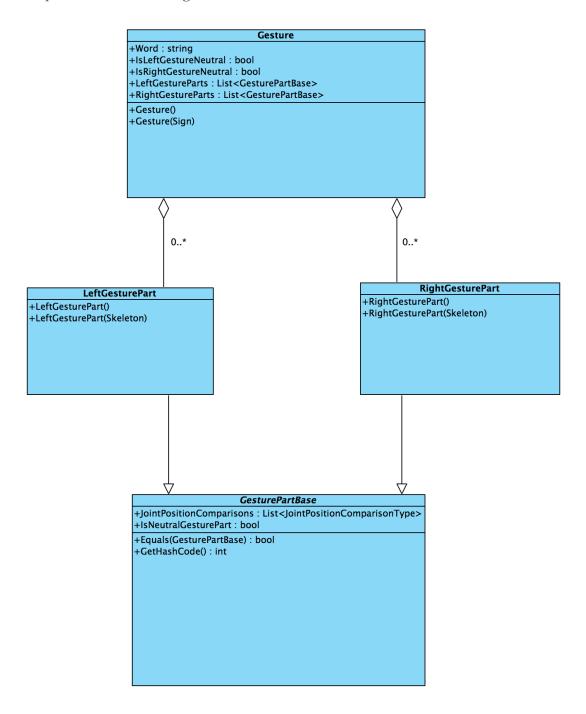
Les signes sont la quintessence des données de TransLSF. Ils sont utilisés dans toutes les parties du projet. C'est eux qui représentent les signes de la langue des signes française avec leur correspondance en mot. Ils peuvent être enregistré, sauvegardé, modifié et supprimé soit dans la base de données, soit dans des fichiers XML.





2. Les gestes

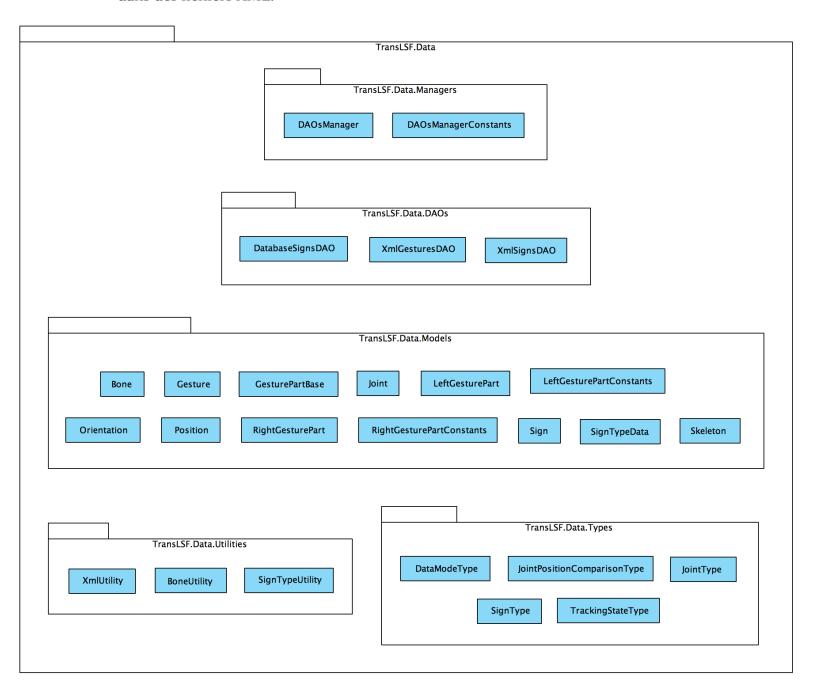
Les gestes sont les objets qui représentent les signes de TransLSF dans le seul but d'être reconnu par l'algorithme de reconnaissance. Ils sont générés par les signes eux-mêmes et sont stockés dans des fichiers XML, pour éviter de les re-transformer à chaque lancement du logiciel.





E. Le gestionnaire de données

La gestion des données est fournit par le DAOsManager qui fait partie du namespace TransLSF.Data.Managers. Il donne accès à un CRUD (Create, Read, Update, Delete) pour les signes et les gestes que ce soit dans la base de données ou dans des fichiers XML.





V. Environnement de développement

A. IDE

TransLSF est développé intégralement grâce à un environnement de développement intégré (IDE) crée par Microsoft : Microsoft Visual Studio, dans sa version 2013.

Que ce soit pour éditer du code en C#, modifier des vues en XAML, modifier des tables dans la base de données, etc... Microsoft Visual Studio s'adapte parfaitement bien au développement de TransLSF.





B. Gestionnaire de version

Le code source de TransLSF est géré et versionné grâce au gestionnaire de version mondialement connu : Git.

Pour gérer les différentes branches du développement, les multiples modifications d'un même code source et les conflits de développement, Git est l'outil le plus approprié.





VI. Documentation du code source

L'ensemble du code source de TransLSF est documenté.

La documentation technique de l'ensemble des classes de TransLSF est accessible à : http://www.translsf.fr/documentation.