

# Sign2Speech

Arthur CLERC-GHERARDI  
Patrick PEREA  
RICM4

# OBJECTIFS DU PROJET

- Reconnaître des gestes du langage des signes
- Afficher la traduction textuelle à l'écran
- Dicter la traduction du geste reconnu

# OBJECTIFS PERSONNELS :

- Reconnaître les gestes en temps réel
- Réaliser un logiciel de type "Skype"

# PLAN

- I. Reconnaissance des gestes
  1. Caractérisation d'un geste
  2. Apprentissage automatique
  3. Synthèse Vocale
- II. Mise en réseau
- III. Démo

# ETAT DE L'ART

- Des ébauches de programmes existent
  - Quelques universités américaines ont déjà tenté un tel projet
- Aucun n'utilise la caméra Intel Creative
  - Technologies utilisées : Kinect avec OpenCV, Zcam, Webcam native
- Des ouvertures de marchés potentiels
  - Intel projette d'équiper ses laptops de la caméra Intel Creative

# RECONNAISSANCE DES GESTES

# CARACTERISATION D'UN GESTE

- Décomposition d'un geste en "symboles"



- Un entier pour représenter un symbole
  - $\approx 32$  bits sur un unsigned int  $\Rightarrow 2^{32}$  possibilités de symboles !
  - Recherche d'un symbole dans le dictionnaire efficace

Bit	0	1	2	3	4	5
Signification	Pouce	Index	Majeur	Annulaire	Auriculaire	Poing fermé

Bit à 1 => Information présente sur le symbole

Bit à 0 => Absence de cette information sur le symbole

Bit 0 à 5 => Doigts de la main

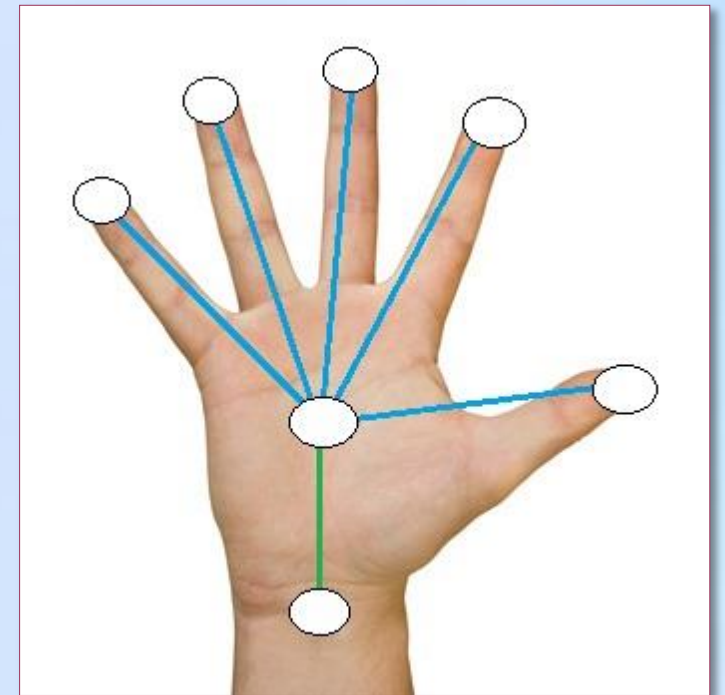
Bit 8 à 15 => Mouvements (gauche, droite, haut, bas ...)

Symbole “main ouverte” = 0x1f



# DETECTION DES DOIGTS

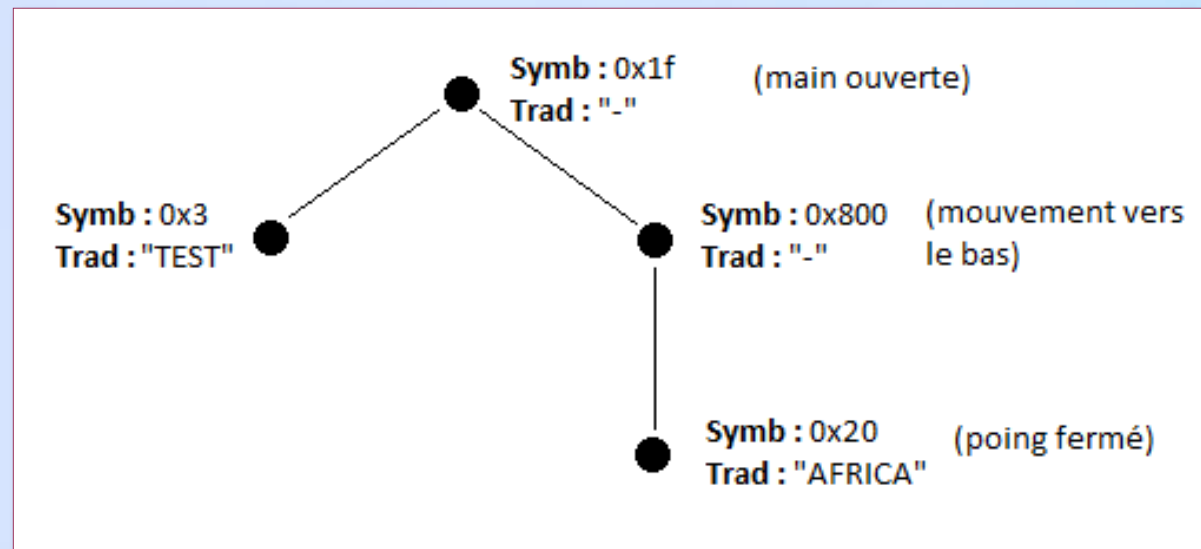
- SDK d'INTEL :
  - Détection de la position des doigts
  - N'indique pas quel est le doigt détecté (pouce...)
- Implémentation d'une surcouche de la SDK
- Elements détectés par la SDK :  
doigts, centre de la main, poignet
- Méthode des angles : Suppose que la main reste face à la caméra et non de profil



# LE DICTIONNAIRE

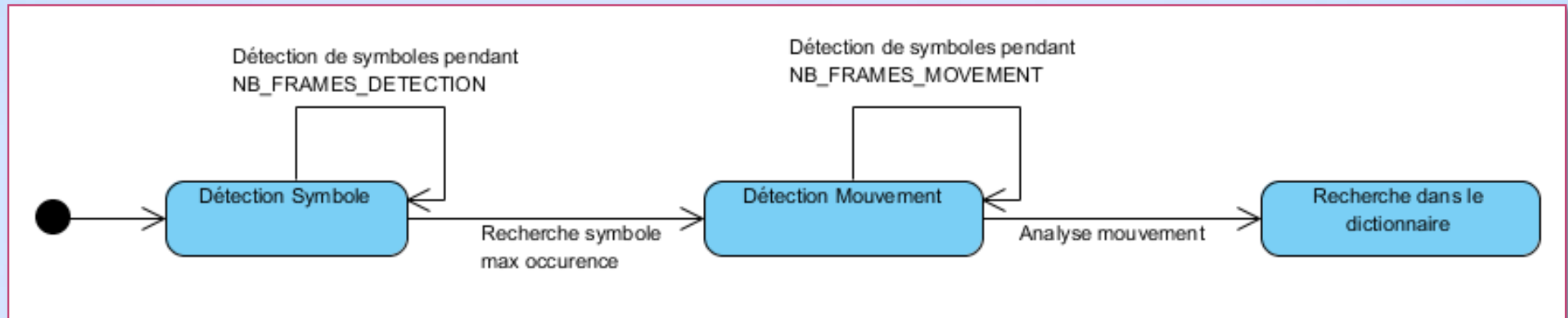
- Contient tous les mots reconnaissable
- Point de départ : fichier XML
- Arbre pour contenir le dictionnaire
  - Permet de “factoriser” un symbole s’il est contenu dans plusieurs gestes

```
<Word name="AIRPLANE">  
  <Symbol value="0x13"/>  
  <Symbol value="0x11413"/>  
</Word>
```



- MECANISME DE RECHERCHE

- Priorité au temps réel



- Recherche rapide et efficace : liste des fils d'un noeud triée par ordre croissant des symboles
    - Recherche dichotomique déjà implémentée

- MOTS RECONNUS :

- ❖ AFRICA

- ❖ AIRPLANE

- ❖ ALL

- ❖ I LOVE YOU

- ❖ UP

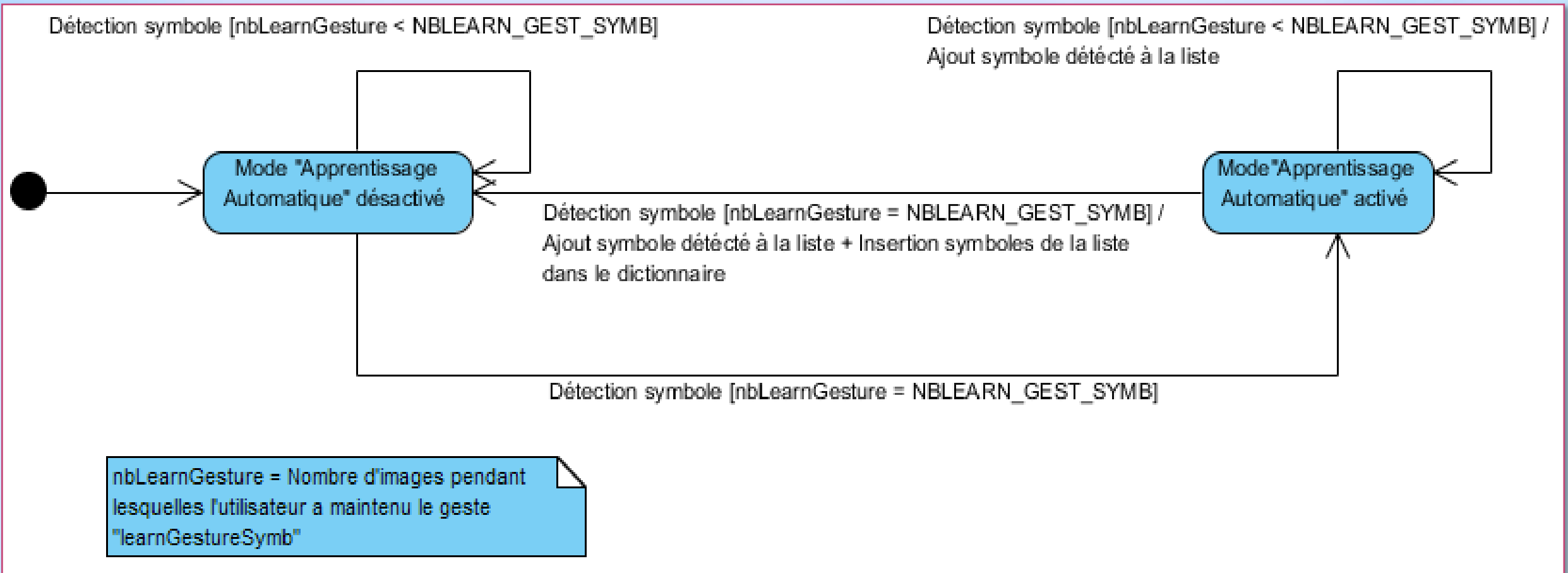
- ❖ ACCIDENT

- ❖ OBSTINATE

Dictionnaire ASL en ligne : <http://www.signingsavvy.com>

# APPRENTISSAGE AUTOMATIQUE

- Un de nos principaux choix de conception :  
L'utilisateur interagit avec l'application en utilisant le plus possible les gestes
- Activation/Désactivation du mode "Apprentissage automatique" :  
Maintien du geste *learnGestureSymb* (main ouverte) pendant *NBLEARN\_GEST\_SYMB*
- Recherche dans l'arbre désactivée pendant l'apprentissage d'un geste
  - Inutile et ralentit l'application



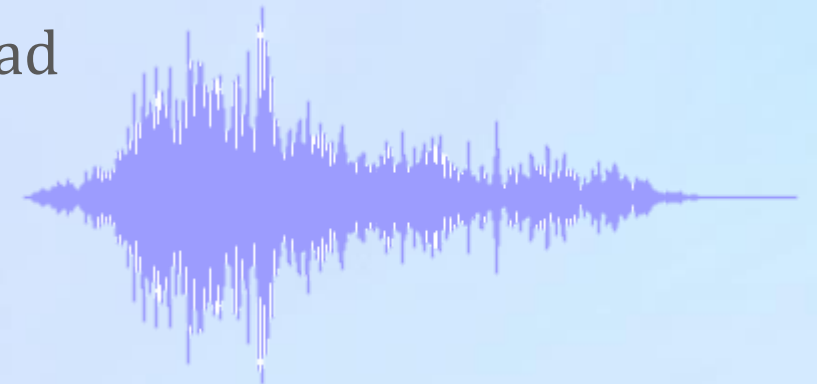
# SYNTHESE VOCALE

- Utilisation de la Microsoft Speech API (SAPI) 5.3
- Simple et en quelques lignes un résultat satisfaisant
- Problème :

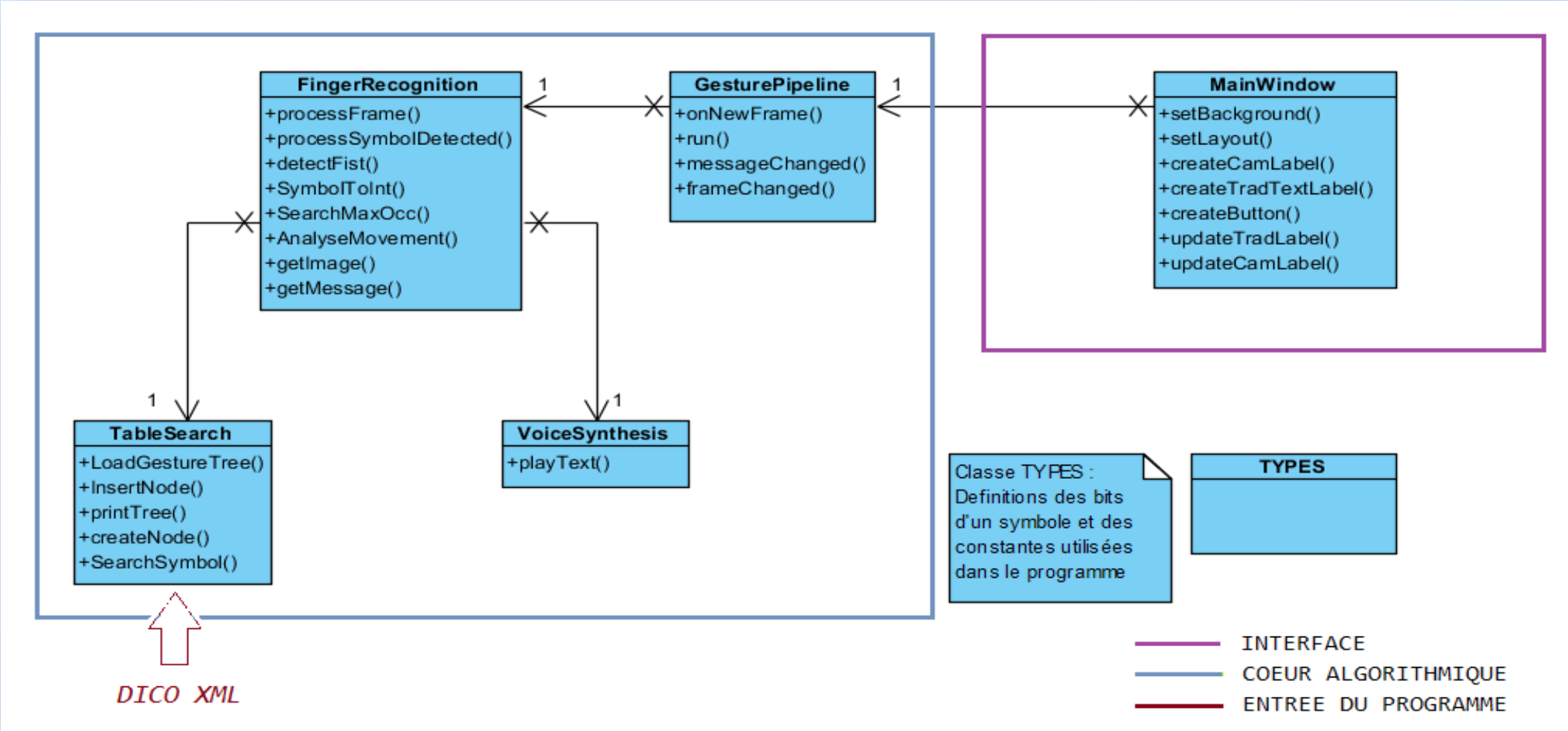
Le stream vidéo se congèle lors de la prononciation d'un mot

- Solution :

Lancer la synthèse vocale dans un autre thread

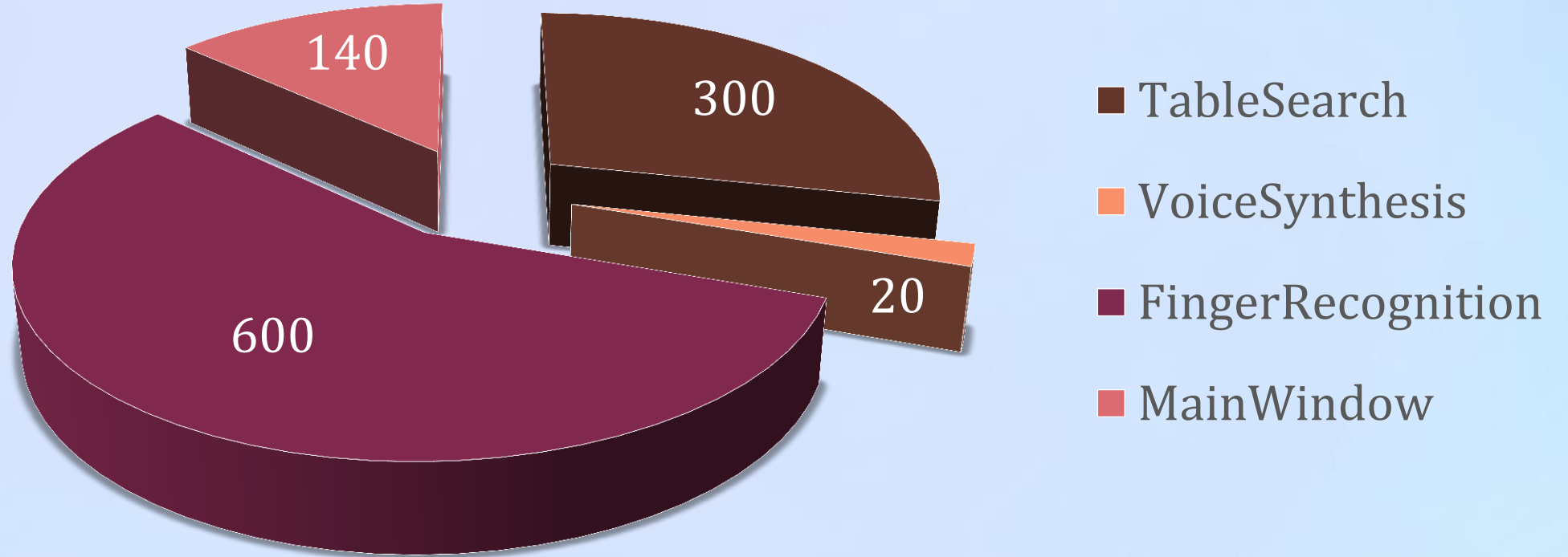


# DIAGRAMME DE CLASSES





# NOMBRE DE LIGNES DE CODE PRODUITES



# MISE EN RESEAU

# CHOIX DE LA TECHNOLOGIE

- Que choisir ?
  - Plugin SKYPE
  - Client – Client puis Multi clients - serveur
- Tentative avec WebRTC
  - Essais concluant avec du Javascript et de l'HTML
  - Intégration en C++ très compliqué
- Souhait d'avoir quelque chose de fonctionnel

# Découverte et essais de transfert de flux vidéo

- Dans un premier temps, client – client en JAVA

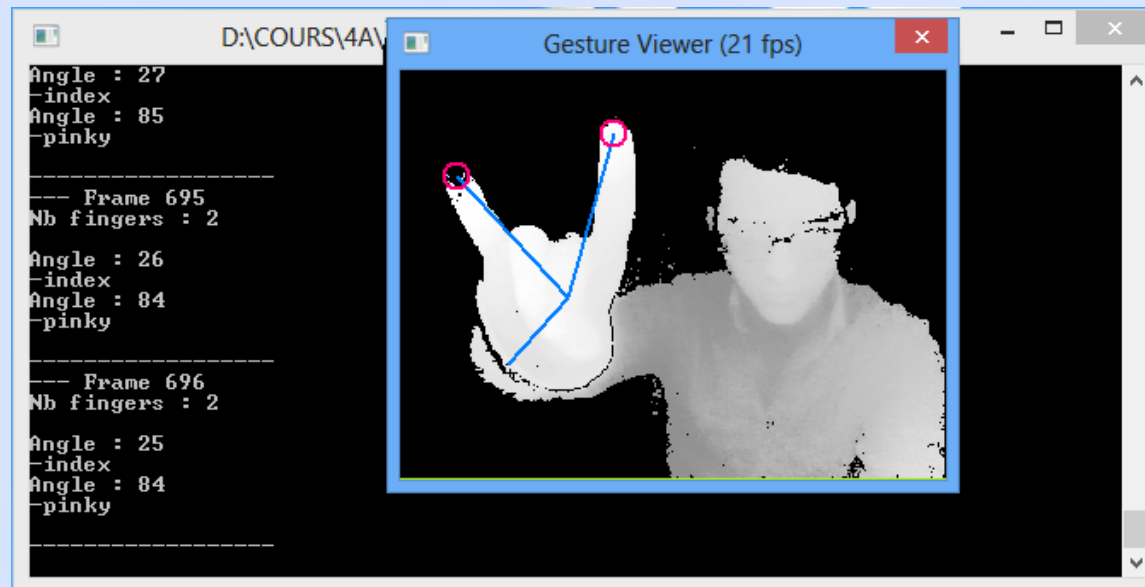
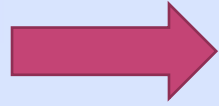


Image BMP en 320 x 240

- Volonté de fluidité dans la conversation et le flux vidéo



20 images par seconde

- Test avec JAVA concluant



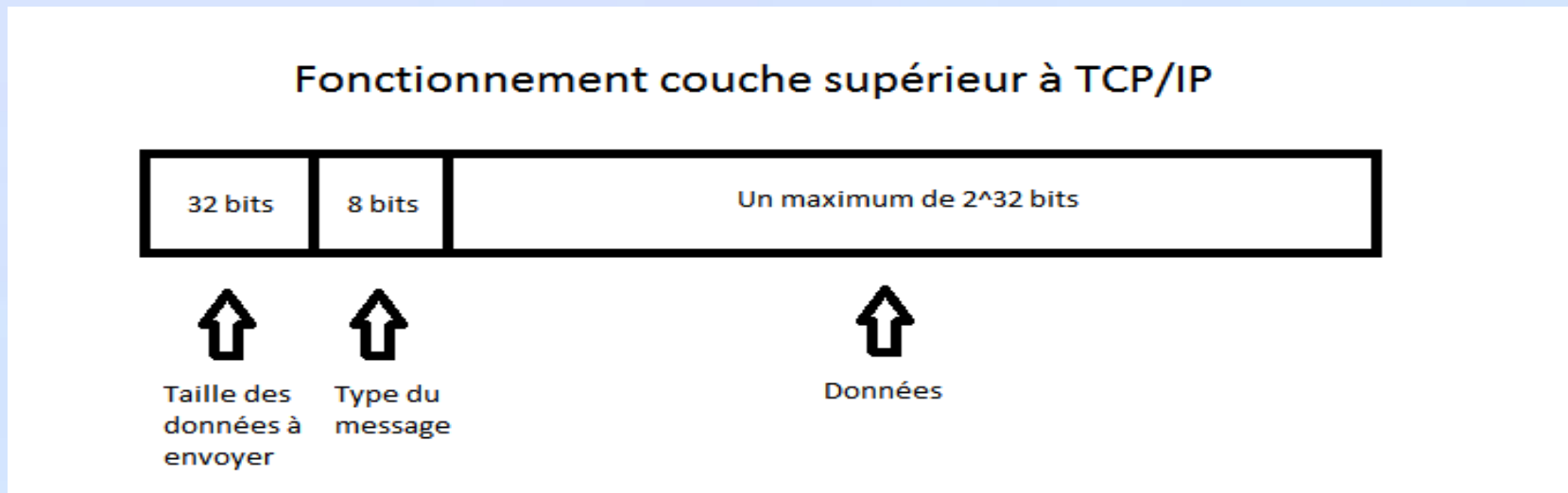
Socket TCP et échange de donnée basique suffisant

- Faire la même chose en C++ pour l'adapter au code de reconnaissance

# ENVOI ET RECEPTION DE DONNEES EN C++

- La sérialisation existe mais moins facilement manipulable qu'en JAVA

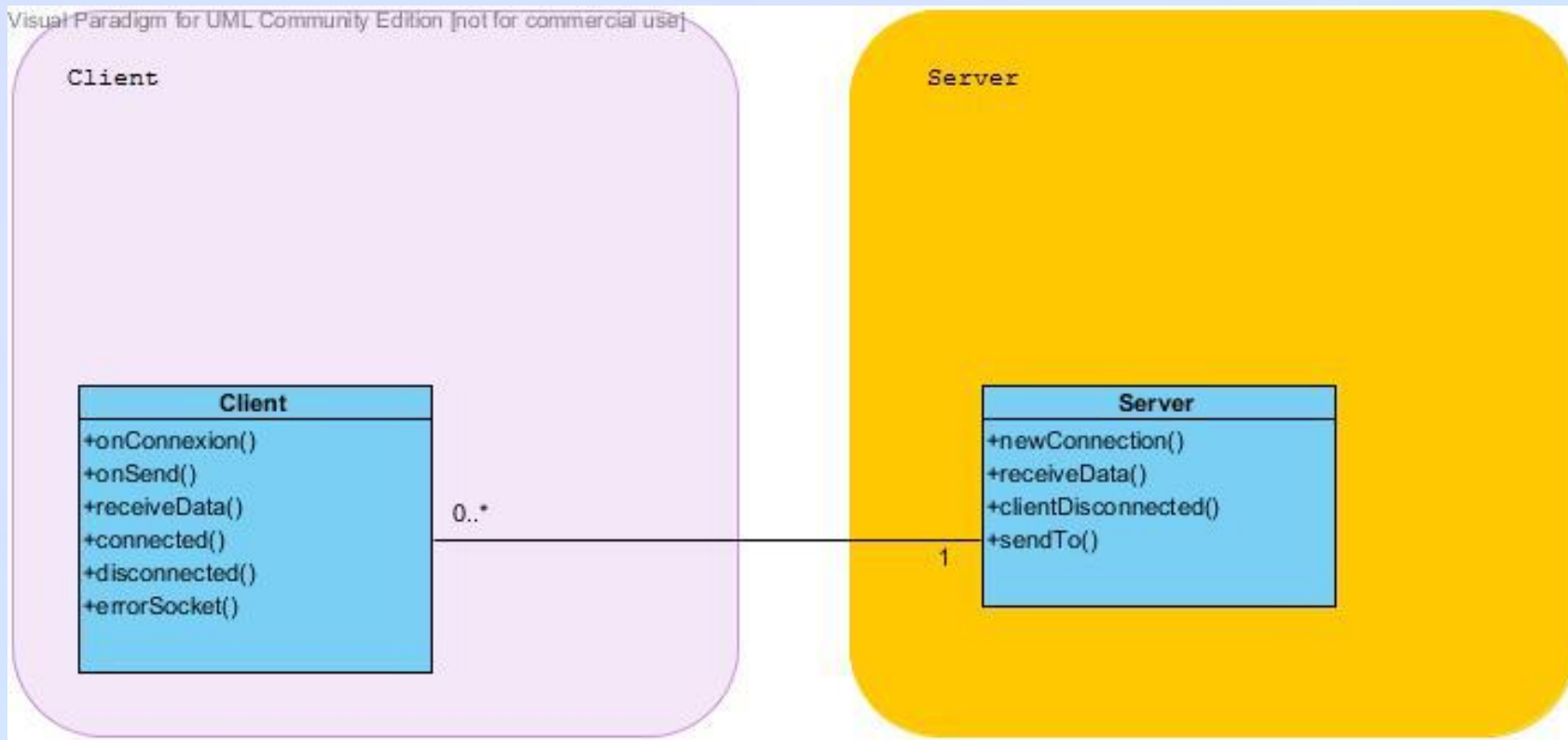
➔ Envoi de données au dessus de la couche TCP/IP



- Volonté de rester dans l'optique d'avoir toujours un code fonctionnel.



On développe pas à pas



# AMELIORATION POSSIBLES

- Si l'échange de données prend trop de bande passante, essayer avec UDP
- WebRTC
- Mise en place de contact en ligne / hors ligne etc ...



# OBJECTIFS ATTEINTS/NON ATTEINTS

# OBJECTIFS ATTEINTS

- ❖ Reconnaissance de gestes en temps réel
- ❖ Affichage de la traduction textuelle
- ❖ Prononciation du mot reconnu
- ❖ Implémentation d'une interface propre et non surchargée

# CE QU'IL RESTE A FAIRE

- ❖ Intégration du module “reconnaissance de gestes” et de du module “mise en réseau”
- ❖ Incorporer plus de mots dans le dictionnaire

# DOCUMENTATION

# Manuel du développeur

Code clair et  
fortement  
commenté

Tutoriel  
permettant le  
chiffrement/  
déchiffrement  
d'un fichier RSA

Page Wiki  
complète et  
lien vers le SRS

# CONCLUSION

- ❖ Projet qui nous a permis de découvrir une nouvelle technologie
- ❖ Découverte/approfondissement des notions en C++
- ❖ Application des principes de génie logiciel
- ❖ Réaliser un logiciel en partant de 0 se révèle très intéressant

# DEMO