

Rapport de projet : Tableau virtuel interactif

Baptiste Saleil

Geoffrey Mélia

Julien Pagès

Kevin Bollini

8 mai 2012

Table des matières

1	Introduction	4
1.1	Présentation	4
1.2	Contexte	4
2	Analyse et conception	5
2.1	Étude de l'existant et faisabilité	5
2.2	Gestion du projet	5
2.2.1	Choix stratégiques	5
2.2.2	Diagramme de Gantt	7
2.3	Outils utilisés	8
2.4	Analyse	9
2.4.1	Cas d'utilisations	9
2.4.2	Diagrammes de séquences	10
2.4.3	Diagramme de classes	12
3	Réalisation	14
3.1	Bibliothèque de suivi	14
3.1.1	15
3.2	Application	15
4	Résultats	16
4.1	Méthodes de suivi	16
4.2	Application	16
5	Conclusion	17
5.1	Difficultés rencontrées	17
5.2	Perspectives	17
5.3	Conclusion	17
6	Références	18
I	Annexes	19
A	Documentation de la librairie	20

Table des figures

2.1	Rétroplanning	7
2.2	Cas d'utilisation dessin	9
2.3	Diagramme de séquence local	10
2.4	Diagramme de séquence réseau	11
2.5	Architecture de la bibliothèque	12
2.6	Diagramme de classe de l'application	13

Remerciements :

Nous tenons à remercier tout particulièrement M. William Puech (enseignant chercheur au LIRMM à l'université Montpellier 2, responsable de la formation du Master informatique IMAGINA) sans qui ce projet n'aurait pas pu se faire, pour avoir accepté de nous encadrer, pour son aide et son implication.

Nous souhaitons aussi remercier M. Benoit Lange (doctorant au LIRMM) pour ses conseils et sa présence au cours de ce projet.

Chapitre 1

Introduction

1.1 Présentation

L'objectif de ce projet est de créer un tableau numérique, sur lequel ou une plusieurs personnes peuvent dessiner en réalisant directement les gestes comme sur une toile réelle.

Pour cela nous utiliserons une interface basée sur la reconnaissance des mouvements détectés par la webcam.

1.2 Contexte

Chapitre 2

Analyse et conception

2.1 Étude de l'existant et faisabilité

Faisabilité

Notre groupe ayant lui-même proposé ce sujet, celui-ci s'accorde parfaitement à nos formations et spécialités. Le choix de ce sujet a donc été réalisé en fonction de nos expériences, savoir-faire et affinités.

Ce projet s'inscrivant dans le cadre d'un TER de notre formation, l'étude de la faisabilité n'inclura pas certains critères comme l'étude de marché, le contexte économique ou le besoin réel. Nous allons nous concentrer notamment sur les compétences techniques et de gestion. Pour les parties financement et analyse des coûts, il va sans dire que nous ne disposons d'aucun financement et n'allons utiliser que des outils gratuits.

Le projet sera décomposé en deux parties majeures que sont la bibliothèque de suivi et l'application de dessin virtuel (dont les contenus seront détaillés dans des sections qui leur sont propres).

Existant

La vision par ordinateur et particulièrement le suivi d'objets, sont des domaines connus et pour lesquels il existe de nombreux travaux. Nous pouvons nous inspirer de certains de ces travaux afin de proposer des fonctionnalités plus pertinentes, éviter certains écueils ou plus directement utiliser des outils existants comme la librairie OpenCV.

- Exemples dans le jeux vidéo : Kinect, Eye-toy, CamSpace (techniques pour l'IHM, par exemple, mouvements continus ou immobilité sur une zone).
- Exemples issus de thèses et de projets de recherche (techniques de programmations).

2.2 Gestion du projet

2.2.1 Choix stratégiques

Lors de ce projet, nous avons distingué deux parties majeures.

La première est une bibliothèque de suivi d'objets. La seconde, une application qui utilise cette bibliothèque pour réaliser un logiciel de dessin interactif.

Ce choix présente plusieurs intérêts :

- Premièrement, ce découpage permet de bien différencier les tâches. Le développement de la librairie requiert des connaissances en traitement d'images pour tracer des couleurs/objets, appliquer des filtres, manipuler les images, etc. Ceci correspond parfaitement à la formation de deux membres du groupe(formation IMAGINA à l'UM2). Le développement de l'application requiert quant à lui des connaissances en génie logiciel, pour l'architecture de l'application, en IHM, pour la réalisation de l'interface (logicielle et gestuelle), ou encore en réseau pour la mise en place de l'architecture client/serveur. Ces différents points correspondent également parfaitement à la formation des deux autres membres du groupe. (Formation AIGLE à l'UM2). Ainsi, les tâches peuvent bien se répartir en fonction des connaissances et spécialités de chacun.
- Deuxièmement, l'intérêt de ce découpage est d'avoir deux projets à la fois complètement indépendants et complémentaires. En effet, la librairie est développée offre des fonctionnalités de suivi utilisables par n'importe quelle application.

L'application se charge d'utiliser les fonctionnalités de traitement d'images de la librairie en ajoutant une interface graphique, une couche réseau, une connexion aux webcams, la récupération des images etc. La librairie et l'application sont donc deux projets développés en parallèle et très modulaires. Notre application peut très bien utiliser une autre librairie de traitement d'images, et la librairie peut très bien être utilisée par d'autres applications qui offriraient des fonctionnalités totalement différentes de la nôtre.

2.2.2 Diagramme de Gantt

Étant donné que nous avons proposé notre propre sujet, la première étape a été de rédiger un cahier des charges pour définir ce que nous voulions faire et de quelle manière.

Après ceci, nous avons commencé à effectuer des recherches du côté bibliothèque et application, pour savoir quels outils et techniques utiliser pour arriver à nos objectifs. Nous avons réservé plusieurs semaines pour effectuer des recherches et perfectionner notre idée du projet. Nous avons donc lu des articles de recherches sur des thématiques proches, effectuer des essais d'outils etc. Cette période d'analyse et de conception sert aussi à penser l'application en imaginant des cas d'utilisations (use cases) et en réalisant le diagramme de classes.

Par la suite, la majeure partie du temps est réservée au développement en lui-même, en prenant en compte la date de présentation du projet pour avoir une réalisation complète.

Bien entendu ces temps de développement s'accompagnent de réunions avec tout le groupe de projet ainsi que de travail pour lier nos deux sous-parties et les tester ensemble.

Des documents comme le rapport ou la documentation furent produit au fur et à mesure du projet, pour rester en adéquation avec le travail accompli. Voici le rétroplanning qui illustre le déroulement chronologique du projet :

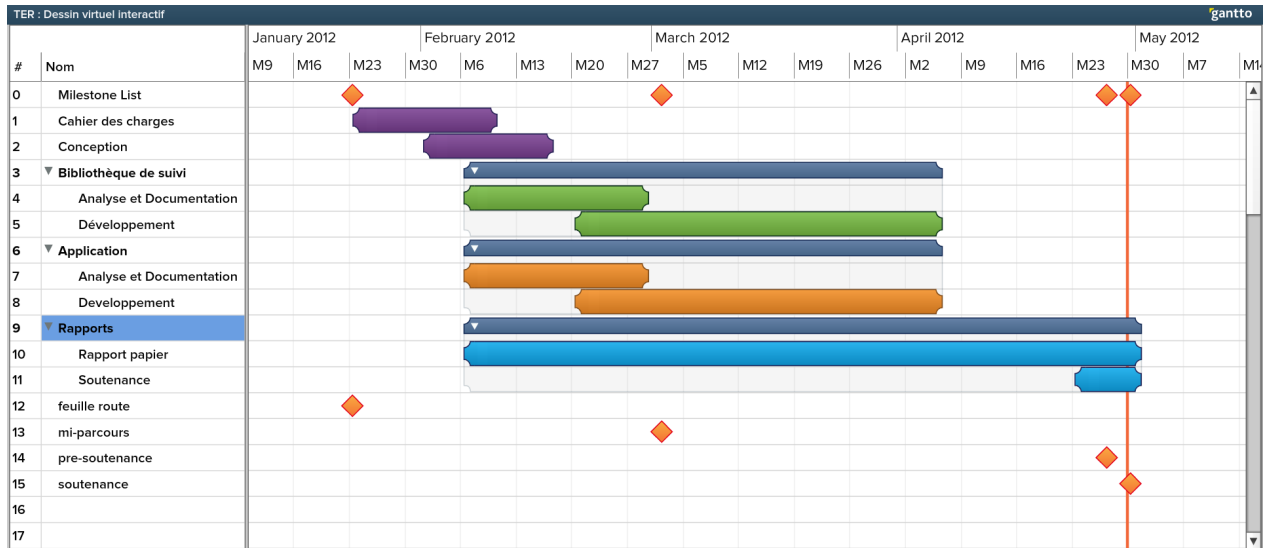


FIGURE 2.1 – Rétroplanning

2.3 Outils utilisés

Durant la période de recherche et au cours du projet nous avons utilisé différents outils (au sens large) pour nous aider dans la réalisation. Nous les détaillons ici dans cette section.

Outils collaboratifs

Tout d'abord nous avons utilisés des outils pour nous aider à synchroniser notre travail entre les deux sous-groupes.



Subversion : Un gestionnaire de versions qui nous permettait de synchroniser notre travail, de partager le code source et de s'échanger quelques documents complexes (ce rapport par exemple).



Gobby : Logiciel libre pour l'édition collaborative, qui nous permet de développer à distance sur le même document simultanément.

Bien sûr ces outils ne sont pas suffisants, nous avons donc fait des réunions régulières pour organiser le travail et décider des orientations, par ailleurs, nous avons fait beaucoup d'échanges pour que tout le monde soit toujours au courant des différentes communications importantes.

Outils techniques

Langage C pour la bibliothèque : ce langage est bien adapté pour écrire une bibliothèque car il nous permet de faire des fonctions, ce qui était dans notre idée ce qu'il fallait pour cette bibliothèque. Par ailleurs c'est un langage bien adapté pour faire du traitement de l'image car il est de bas niveau, et très rapide. De plus nous avons utilisé par la suite des bibliothèques (OpenCV par exemple) qui ont été écrites initialement en C.

Langage C++ pour l'application : c'est un langage objet ce qui nous aidait pour la conception, il est également très rapide et c'est un langage avec lequel nous sommes à l'aise et nous avons de l'expérience, ça nous évitait donc de devoir réapprendre et donc de perdre du temps précieux.

Pour ce projet nous utilisons plusieurs bibliothèques pour nous aider :



Qt : bibliothèque d'IHM (et bien plus) en C++, nous aide pour faire les interfaces graphiques, pour la partie réseau également où elle offre des abstractions agréables.



OpenCV : bibliothèque de traitement de l'image, nous aide à exploiter la webcam et pour le traitement de l'image avec plusieurs fonctionnalités.

2.4 Analyse

2.4.1 Cas d'utilisations

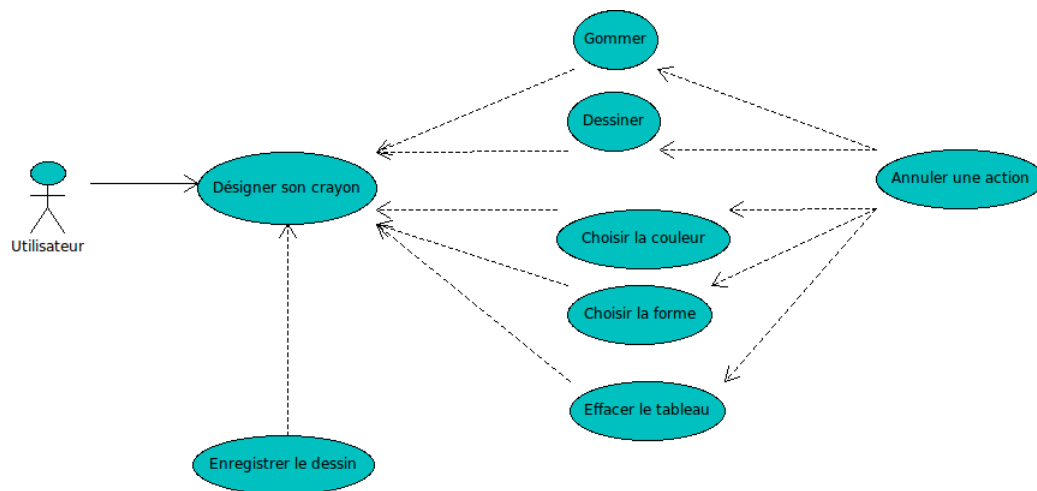


FIGURE 2.2 – Cas d'utilisation dessin

2.4.2 Diagrammes de séquences

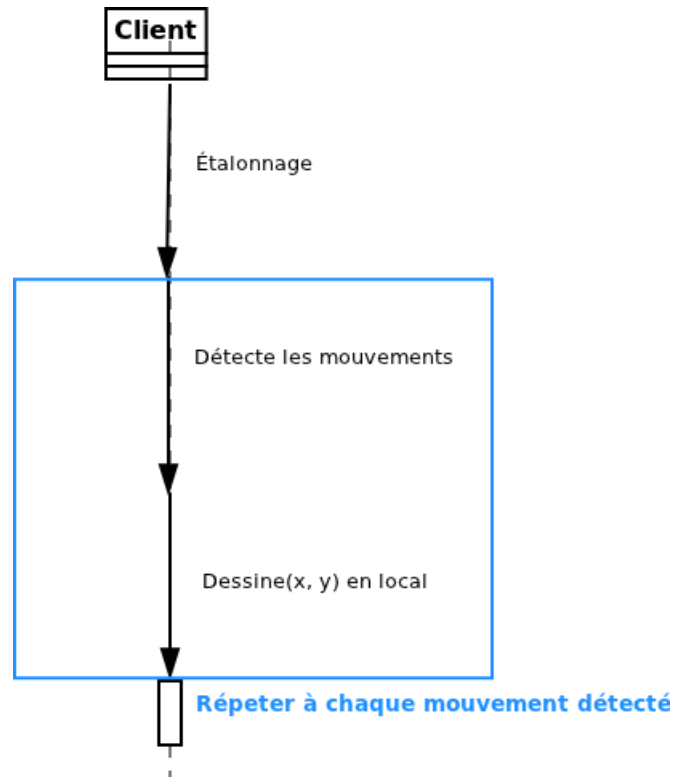
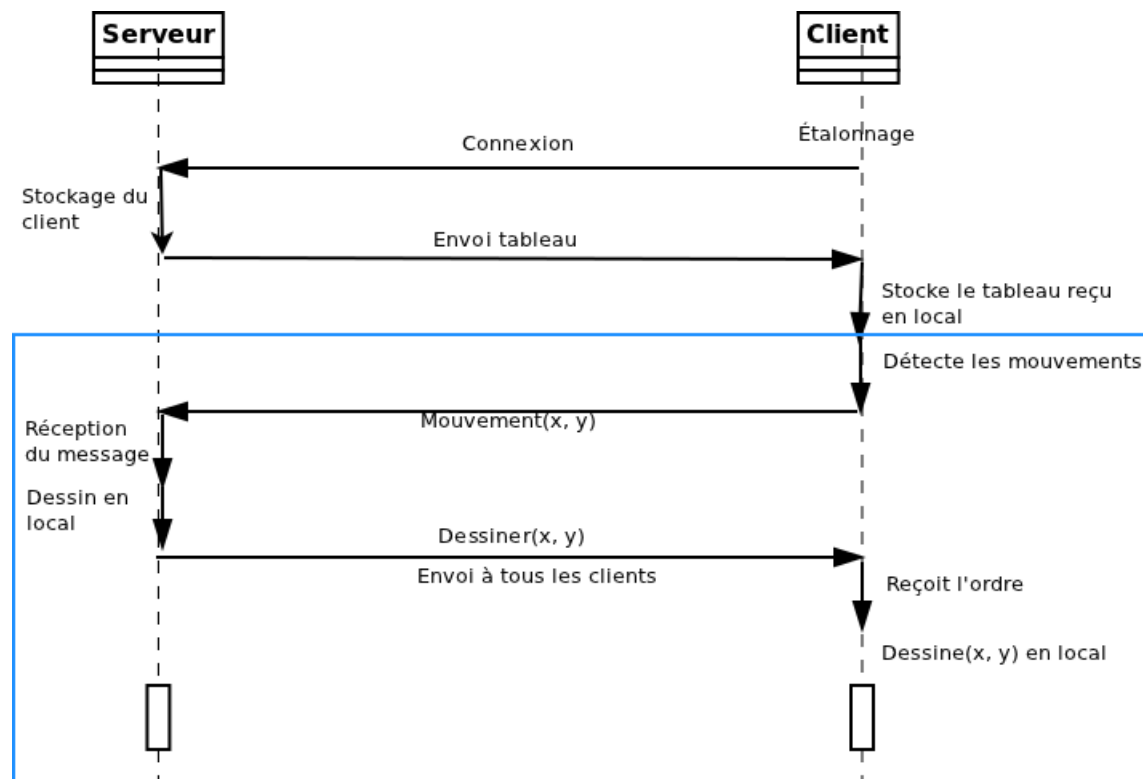


FIGURE 2.3 – Diagramme de séquence local



Répéter à chaque mouvement détecté

FIGURE 2.4 – Diagramme de séquence réseau

2.4.3 Diagramme de classes

La bibliothèque fonctionne grâce à une structure de données utilisée par deux fonctions enveloppes. Le curseur est représenté en mémoire sous la forme d'une structure de données "Cursor". Cette structure de données est engendrée par la fonction "calibration" de la librairie. Elle est ensuite utilisée par la fonction "track", qui permet la mise à jour des informations relatives au curseur par rapport à une image.

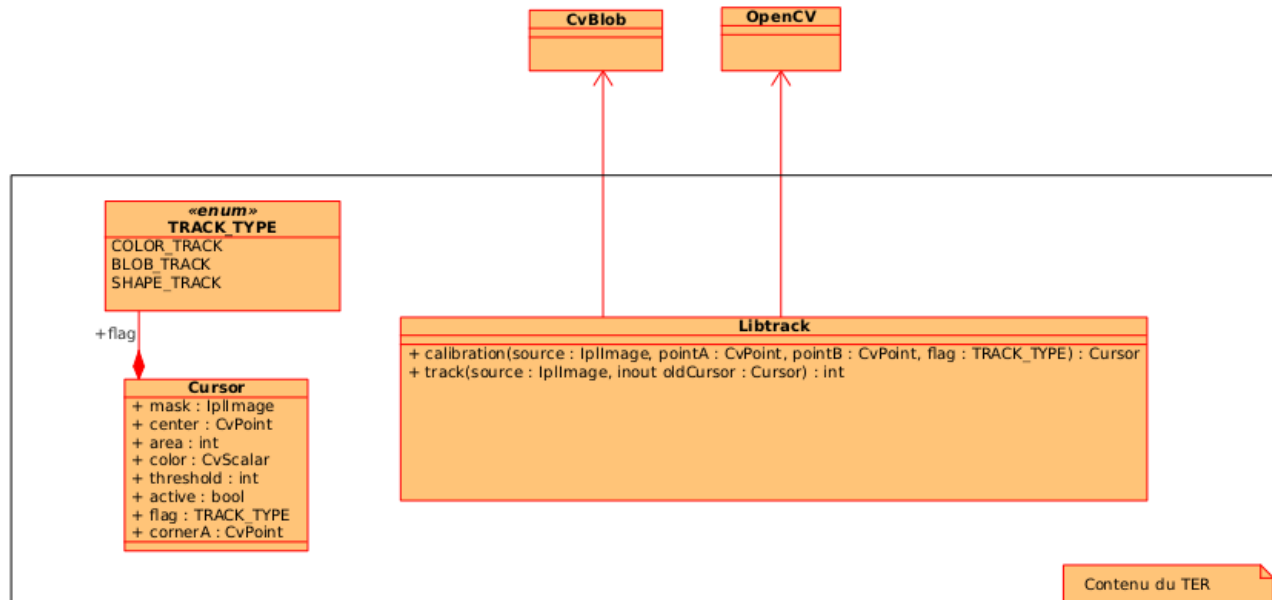


FIGURE 2.5 – Architecture de la bibliothèque

La bibliothèque dépend de deux bibliothèques externes :

- OpenCV, Spécialisée dans le traitement des images
- CvBlob, librairie issue d'OpenCV utilisée pour la recherche d'objet dans des images binaires

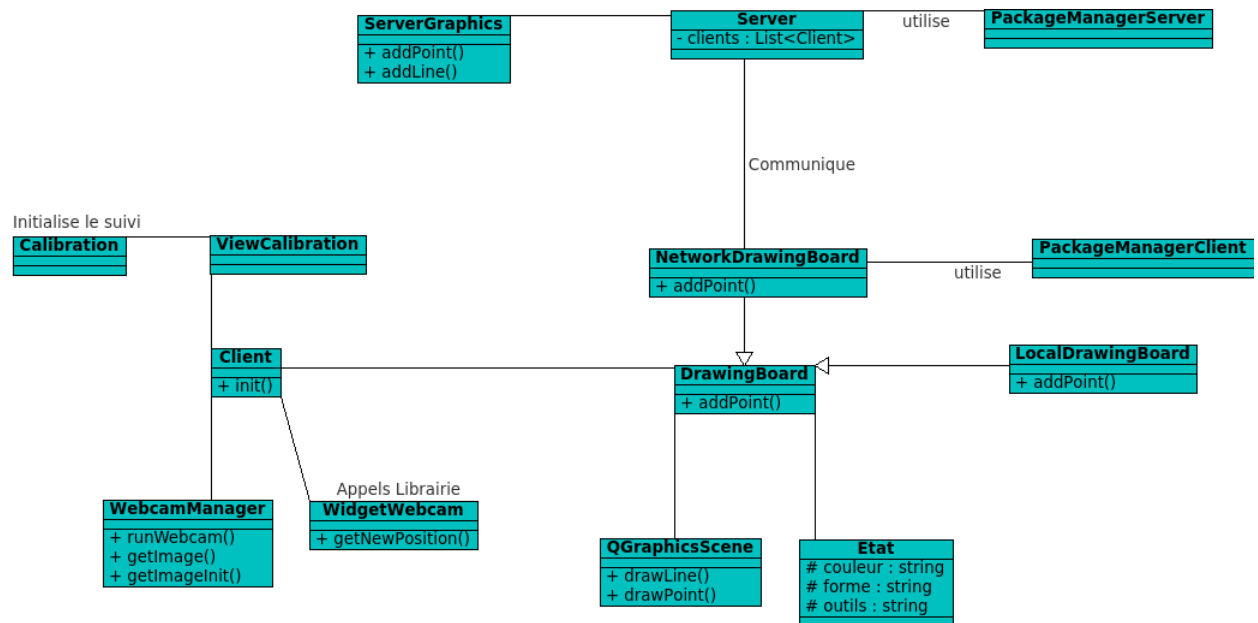


FIGURE 2.6 – Diagramme de classe de l'application

Chapitre 3

Réalisation

3.1 Bibliothèque de suivi

Cette partie du développement a pour objectif de fournir un ensemble de fonctionnalités permettant la reconnaissance et le suivi d'un objet, que nous appellerons ici Curseur, à partir d'images.

L'objectif est de développer notre propre bibliothèque pour d'une part avoir un outil remplissant précisément nos objectifs et d'autre part de faire un travail de recherche et développement dans le domaine du traitement de l'image (en particulier le suivi d'objets).

Un certain nombre de bibliothèques existantes proposent des moyens nous facilitant la tâche, notamment grâce à des outils de reconnaissance de formes, de suivis de modèle ou d'évaluation de composantes connexes. Le travail consiste donc à tirer partie de ces bibliothèques en les associant de manière à proposer des méthodes efficaces et simples à utiliser. Cela implique une bonne connaissance des outils existants, de leur compréhension et de leur utilisation conjointes dans le but de faire notre propre bibliothèque.

La Bibliothèque de suivi sert à réaliser le suivi d'un objet, que nous appellerons ici Curseur, avec un minimum de connaissance en traitement des images.

La bibliothèque fonctionne grâce à une structure de données manipulée par deux fonctions enveloppes.

Le curseur est représenté en mémoire sous la forme d'une structure de données "Cursor".

Cette structure de données est engendrée par la fonction "calibration" de la librairie. Elle est ensuite utilisée par la fonction "track", qui permet la mise à jour des informations relatives au curseur.

Structure de données

Fonctions

3.1.1

3.2 Application

Application :

L'application doit exploiter le plus efficacement possible la librairie de reconnaissance. Nous voulons donc mettre en place un étalonnage de l'objet à suivre le plus précisément possible. Le projet doit ensuite être exploitable et utile, nous devons donc mettre en place un certain nombre d'outils tels que la possibilité de gommer, changer de forme, de couleur etc. Un module réseau est aussi prévu pour une utilisation concurrente et synchronisée. Et bien sûr la fonctionnalité clé de l'application est le tableau virtuel, qui devra donc être exploitable et le dessin devra être fidèle aux mouvements reconnus.

Chapitre 4

Résultats

4.1 Méthodes de suivi

4.2 Application

Chapitre 5

Conclusion

5.1 Difficultés rencontrées

5.2 Perspectives

5.3 Conclusion

Chapitre 6

Références

Première partie

Annexes

Annexe A

Documentation de la librairie