

# Rapport de projet : Tableau virtuel interactif

Baptiste Saleil

Geoffrey Mélia

Julien Pagès

Kevin Bollini

4 mai 2012

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Présentation . . . . .	3
1.2	Contexte . . . . .	3
<b>2</b>	<b>Analyse et conception</b>	<b>4</b>
2.1	Étude de l'existant et faisabilité . . . . .	4
2.2	Gestion du projet . . . . .	4
2.2.1	Choix stratégiques . . . . .	4
2.2.2	Diagramme de Gantt . . . . .	6
2.3	Outils utilisés . . . . .	7
2.4	Analyse . . . . .	8
2.4.1	Cas d'utilisations . . . . .	8
2.4.2	Diagrammes de séquences . . . . .	9
2.4.3	Diagramme de classes . . . . .	11
<b>3</b>	<b>Réalisation</b>	<b>13</b>
3.1	Bibliothèque de suivi . . . . .	13
3.1.1	. . . . .	14
3.2	Application . . . . .	14
<b>4</b>	<b>Résultats</b>	<b>15</b>
4.1	Méthodes de suivi . . . . .	15
4.2	Application . . . . .	15
<b>5</b>	<b>Conclusion</b>	<b>16</b>
5.1	Difficultés rencontrées . . . . .	16
5.2	Perspectives . . . . .	16
5.3	Conclusion . . . . .	16
<b>6</b>	<b>Références</b>	<b>17</b>
<b>I</b>	<b>Annexes</b>	<b>18</b>
<b>A</b>	<b>Documentation de la librairie</b>	<b>19</b>

# Table des figures

2.1	Retroplanning . . . . .	6
2.2	Cas d'utilisation dessin . . . . .	8
2.3	Diagramme de séquence local . . . . .	9
2.4	Diagramme de séquence réseau . . . . .	10
2.5	Architecture de la bibliothèque . . . . .	11
2.6	Diagramme de classe de l'application . . . . .	12

# Chapitre 1

## Introduction

### 1.1 Présentation

Le but principal du projet est de simuler une écriture ou un dessin sur un tableau virtuel interactif. Pour cela nous utiliserons une interface basée sur la reconnaissance de mouvements.

### 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).

#### 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.

#### 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

Pour mener à bien ce projet nous avons décidés de découper les tâches en deux sous-parties.

La première sera une bibliothèque de suivi d'objets, et la deuxième partie une application qui utilisera 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 Université Montpellier

2). Le développement de l'application, lui, requiert 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 Université Montpellier 2). 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 complètement indépendants. En effet, la librairie sera développée d'un côté et permettra d'offrir un bon nombre de fonctionnalités de suivis. L'application, elle, se chargera d'utiliser les fonctionnalités de traitement d'image 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 parallèles 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 application qui offriraient des fonctionnalités totalement différentes de la nôtre.

## 2.2.2 Diagramme de Gantt

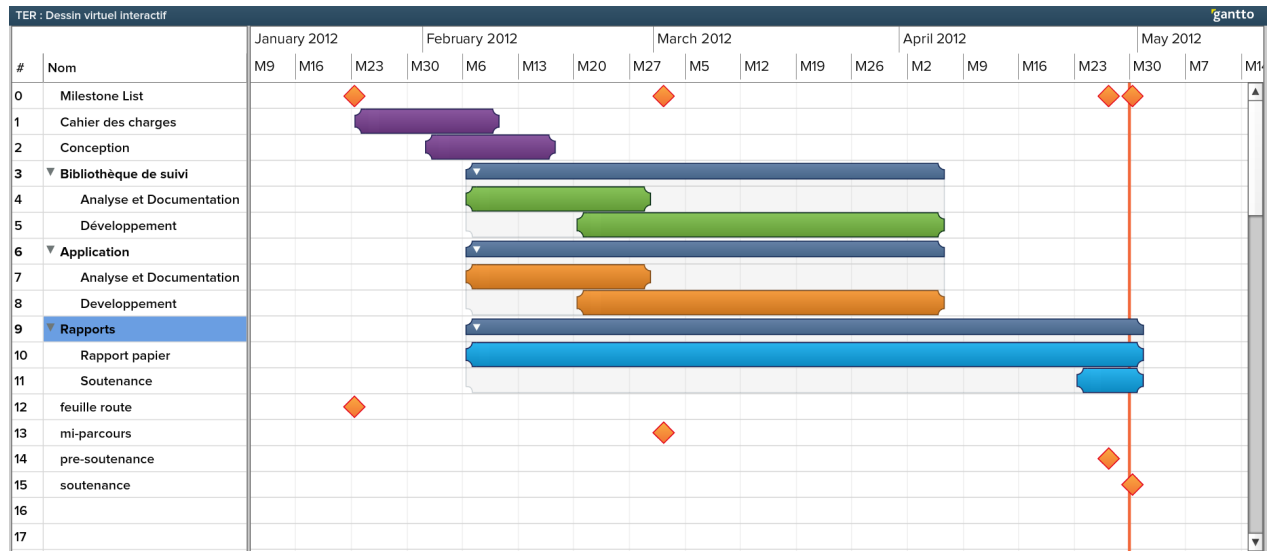


FIGURE 2.1 – Retroplanning

## 2.3 Outils utilisés



## 2.4 Analyse

### 2.4.1 Cas d'utilisations

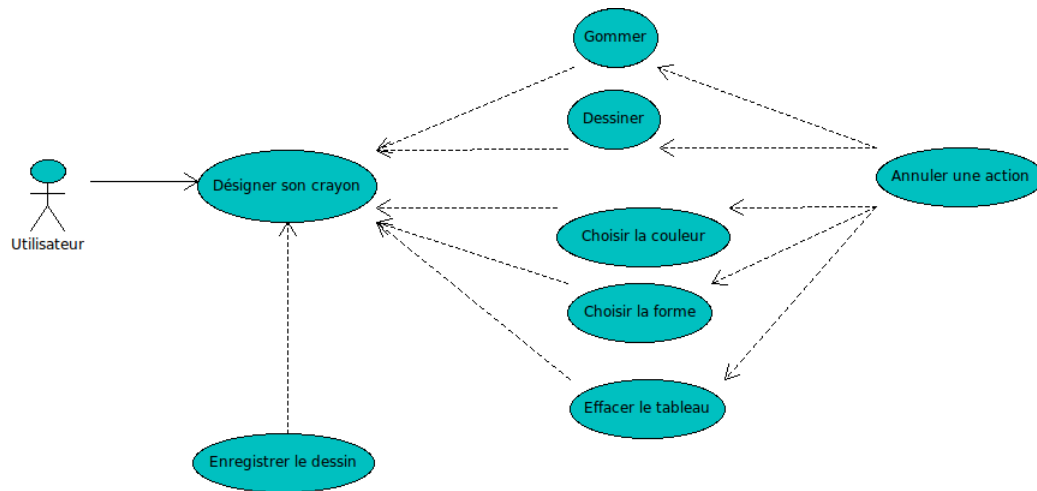


FIGURE 2.2 – Cas d'utilisation dessin

### 2.4.2 Diagrammes de séquences

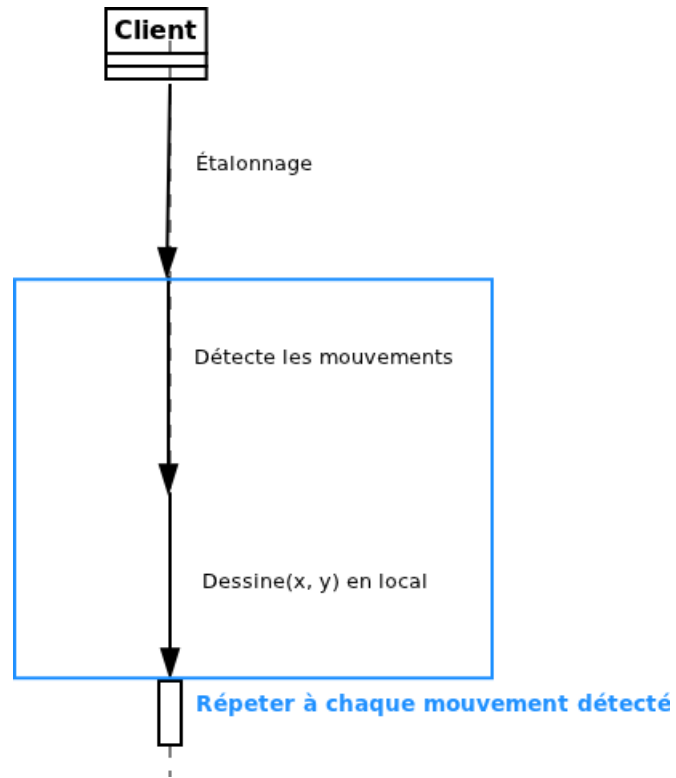
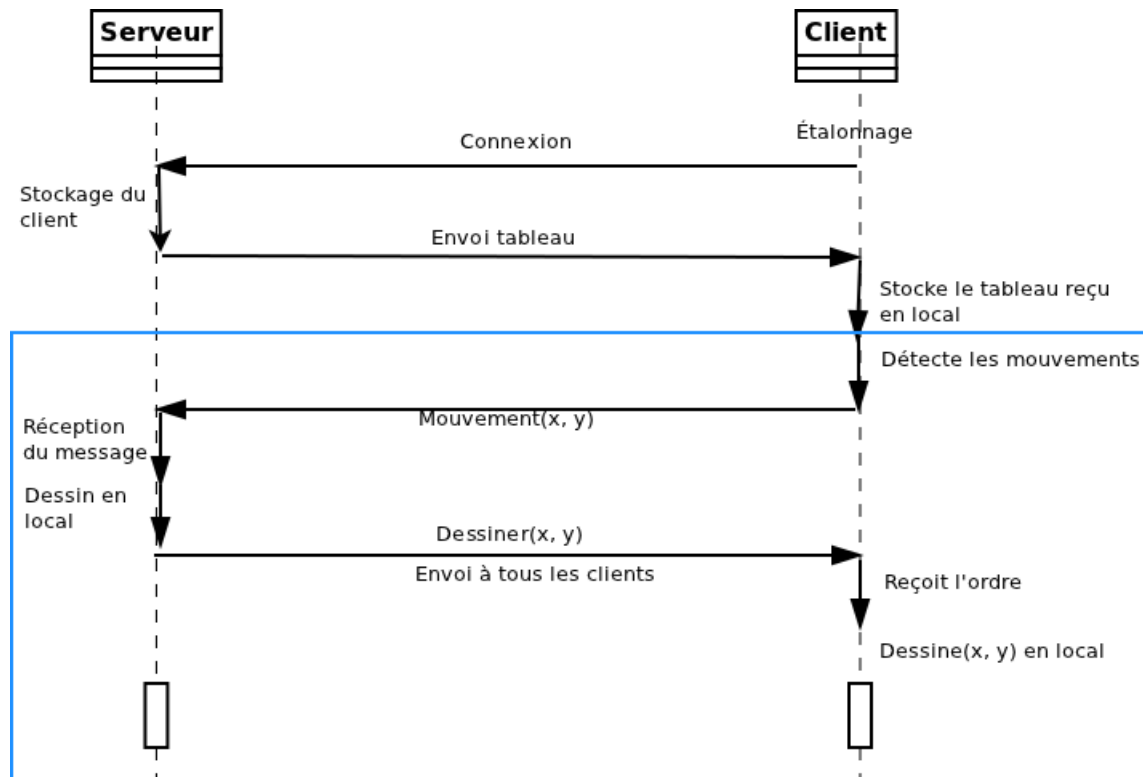


FIGURE 2.3 – Diagramme de séquence local



**Répéter à chaque mouvement détecté**

FIGURE 2.4 – Diagramme de séquence réseau

### 2.4.3 Diagramme de classes

La bibliothèque fonctionne grâce à deux choses, une structure de données et deux fonctions enveloppe. 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, et est utilisée par la suite par l'autre fonction, la fonction "track", qui permet la mise à jour des informations relatives au curseur par rapport à une image.

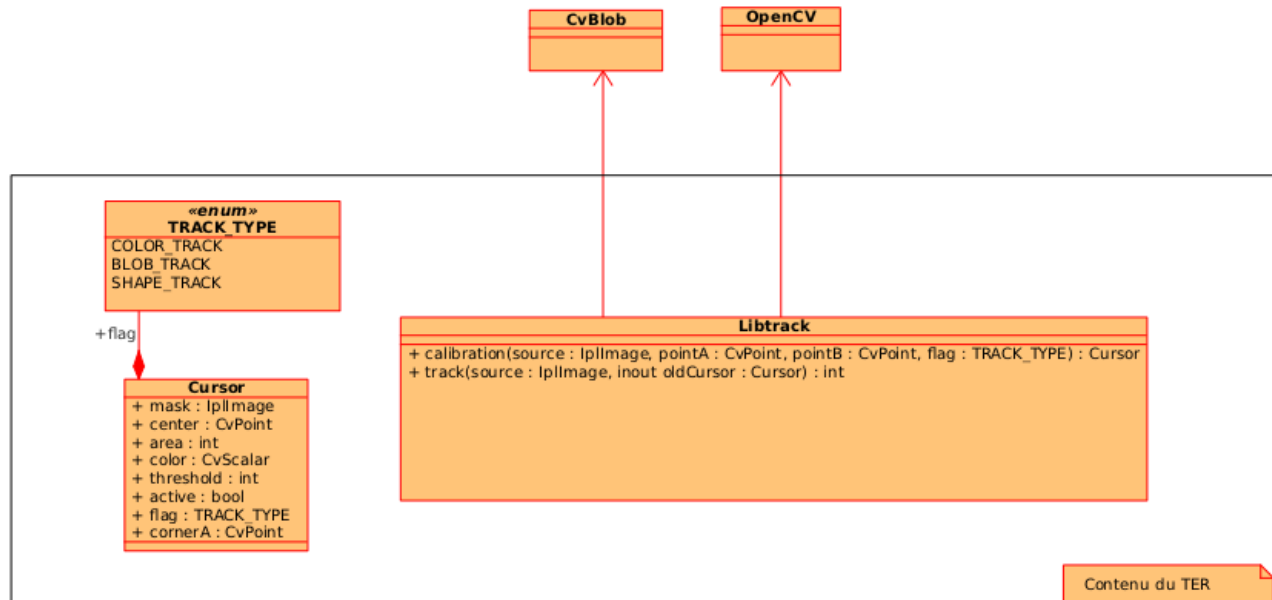


FIGURE 2.5 – Architecture de la bibliothèque

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

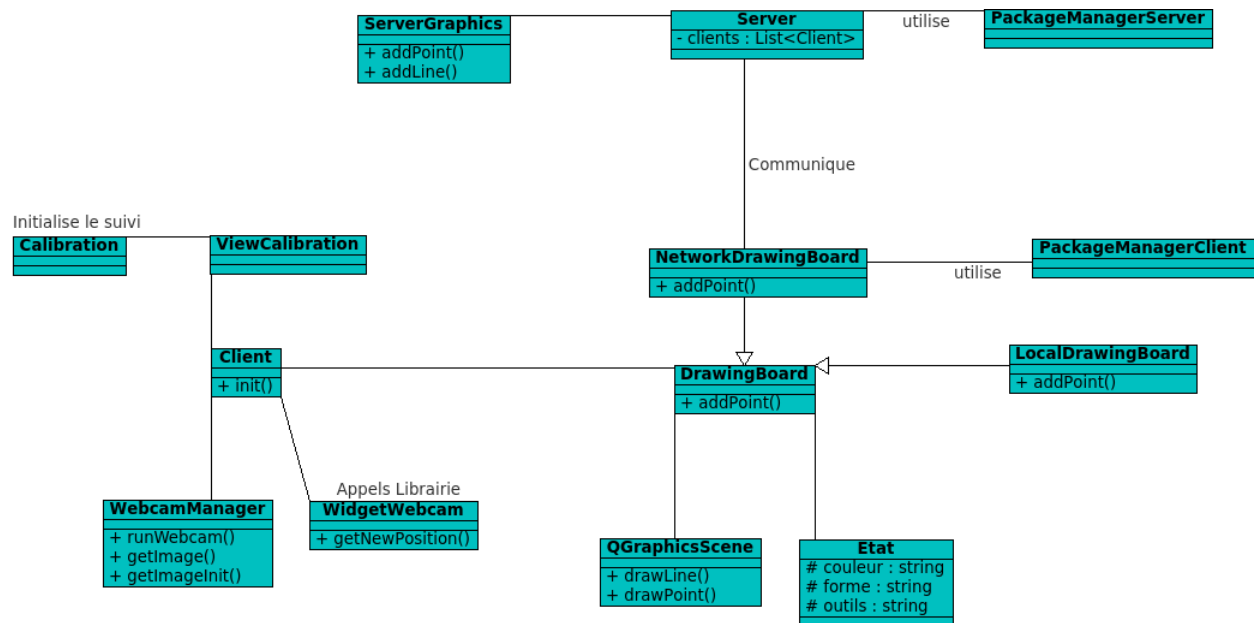


FIGURE 2.6 – Diagramme de classe de l'application

## 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 deux choses, une structure de données et deux fonctions enveloppe.

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, et est utilisée par la suite par l'autre fonction, la fonction "track", qui permet la mise à jour des informations relatives au curseur par rapport à une image.

**Structure de données**

**Fonctions**

**3.1.1**

**3.2 Application**

## **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**