

# Tableau virtuel interactif

Baptiste Saleil, Geoffrey Mélia, Julien Pagès, Kevin Bollini

25 avril 2012

## 1 Introduction

## 2 Analyse et Conception

- Choix de conceptions
- Gestion de projet
- Analyse
- Planning

## 3 Librairie

- Suivi par Couleur
- Suivi par Forme
- Comparatifs

## 4 Application

- Module Local
- Module Réseau

## 5 Conclusion

- 1 Introduction
- 2 Analyse et Conception
- 3 Librairie
- 4 Application
- 5 Conclusion

## Ter de Master 1 : Tableau virtuel interactif

Tuteur : W. Puech

But du projet :

- Concevoir une application avec une interface naturelle (mouvements)
- Librairie de reconnaissance de mouvements
- Application pour exploiter cette librairie pour dessiner ou écrire

- 1 Introduction
- 2 Analyse et Conception**
- 3 Librairie
- 4 Application
- 5 Conclusion

## Choix principaux

Découper le projet en deux parties distinctes :

- une librairie réutilisable
- une application avec une interface naturelle exploitant cette librairie

Objectif :

- Se renseigner, réaliser une architecture de qualité
- Travailler en groupe suivant les compétences
- Développer rapidement un prototype
- Développement incrémental en ajoutant des fonctionnalités

## Organisation :

- Réunions
- Deux sous-groupes
- Partage des tâches au sein des groupes
- Décisions communes (à quatre)

## Collaboration :

- Gestionnaire de version (Subversion)
- Partage de documents (Mail et Subversion)
- Discussions (Mails / Instantanée)
- Édition collaborative pour le travail à distance (Gobby)

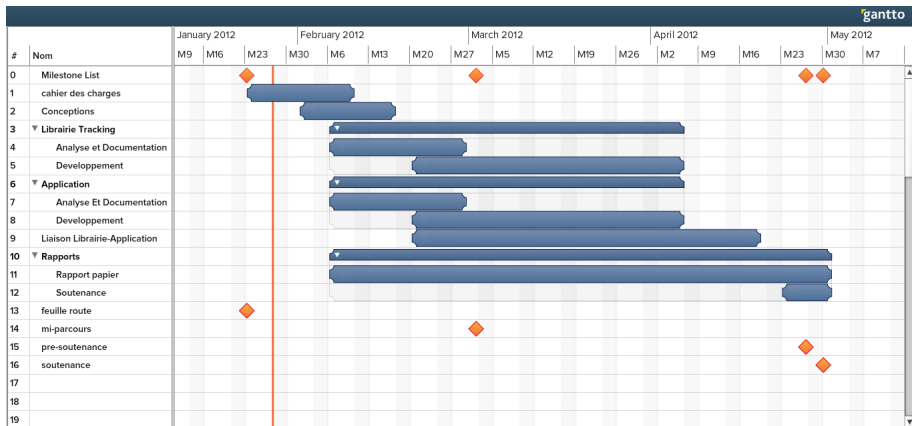


## Objectifs

- Identifier les besoins et envies des utilisateurs
- Distinguer et classer les fonctionnalités de l'application
- Établir un schéma de conception dans le temps
- Faciliter le développement, avoir des buts concrets
- Produire une application réellement aboutie

# Rétroplanning

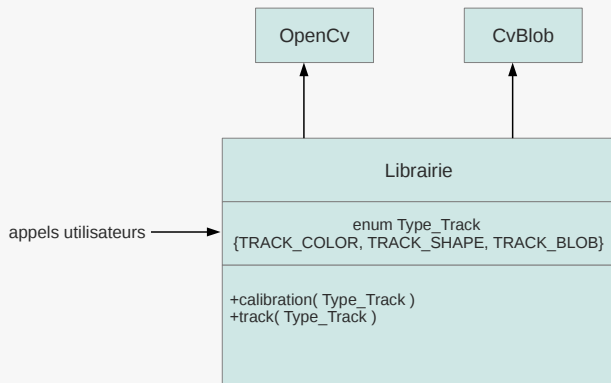
Rétroplanning (Diagramme de gantt) :



- 1 Introduction
- 2 Analyse et Conception
- 3 Librairie**
- 4 Application
- 5 Conclusion

## Objectifs de la librairie conçue

- Distinguer complètement le suivi d'objet de l'application
- Avoir une utilisation simple sans connaissance en traitement d'image
- Permettre une détection d'action
- Proposer un maximum de solutions de traçage
- Évaluer et comparer ces solutions



→ Utilise

Création d'une structure de données : Cursor

```
struct Cursor {
```

- CvPoint center
- CvPoint cornerA
- CvPoint cornerB
- ...
- IplImage \*mask
- Bool active

```
}
```

Deux fonctions enveloppes :

- `Cursor * calibration(IplImage * source, CvPoint A, CvPoint B, TYPE-TRACK flag)`
- `int track(IplImage * source, Cursor * oldCursor)`

Deux types de suivis ont été développés :

- Traçage par couleur
- Traçage par forme



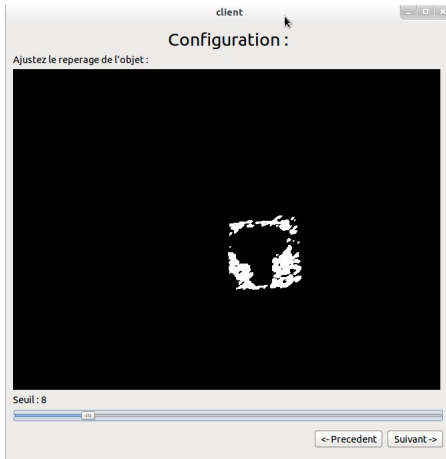
- Sélection de l'objet
- Détection de couleur
- Réglage du seuil de la binarisation
- Binarisation selon le seuil

Deux méthodes de détection de position :

- Calcul du barycentre de l'image binaire
- Recherche par composantes connexes sur l'image

# Suivi par couleur : Barycentre

- Calcul du barycentre de l'image binaire



- Calcul des différentes composantes connexes sur l'image binaire
- Sélection de la composante correspondant au curseur
- Récupération de son barycentre

- Sélection de la zone-objet
- Création d'une sous-image Template

- Recherche du template dans l'image
- Sélection de la zone avec la meilleure correspondance

# Comparatif Couleur/Forme

## Couleur

### Avantages

- Traçage rapide
- Diversité possible de curseurs

### Faiblesses

- Sensibilité à l'environnement
- Dépendant de la qualité du dispositif d'acquisition



# Comparatif Couleur/Forme

## Couleur

### Avantages

- Traçage rapide
- Diversité possible de curseurs

### Faiblesses

- Sensibilité à l'environnement
- Dépendant de la qualité du dispositif d'acquisition

## Forme

### Avantages

- Traçage moins dépendant de la qualité de l'environnement
- Efficace sur des objets 'complexes'

### Faiblesses

- Traçage lent
- Très sensible aux variations du curseur

# Comparatif Simple/composante connexe

## Barycentre simple

### Avantages

- Traçage rapide

### Faiblesses

- Sensibilité aux parasites (fausses détections)
- Précision fortement dépendante de l'environnement

# Comparatif Simple/composante connexe

## Barycentre simple

### Avantages

- Traçage rapide

### Faiblesses

- Sensibilité aux parasites (fausses détections)
- Précision fortement dépendante de l'environnement

## Barycentre composante connexe

### Avantages

- Traçage plus précis
- Résistance aux parasites

### Faiblesses

- Traçage plus lent
- Perte occasionnelle du curseur

## Objectifs atteints

- Librairie utilisable proposant plusieurs solution de traçage
- Détection d'action implémentée dans deux des trois solutions
- Utilisation simple sans connaissances en traitement des images

## Objectifs atteints

- Librairie utilisable proposant plusieurs solution de traçage
- Détection d'action implémentée dans deux des trois solutions
- Utilisation simple sans connaissances en traitement des images

## Difficultés et ouverture

- Exploitation des librairies OpenCv et CvBlob
- Implémentation de la détection pour le suivi par forme
- Ajout à la librairie de nouvelles fonctions de suivi

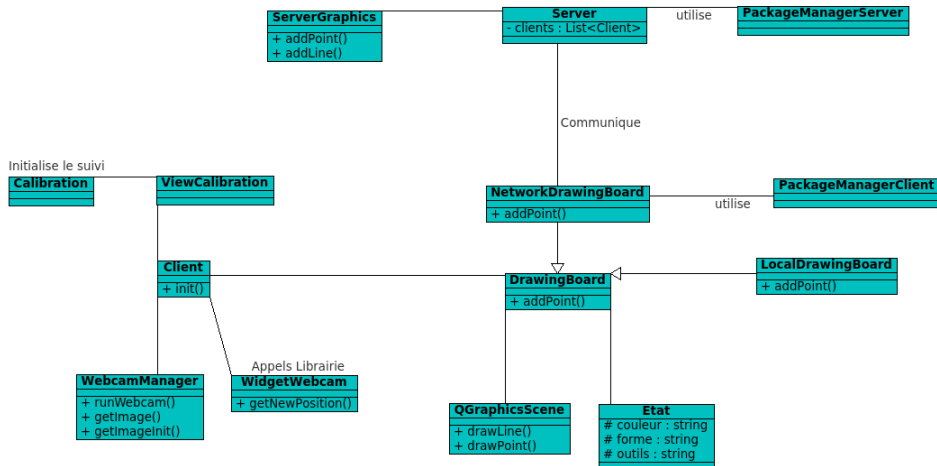
- 1 Introduction
- 2 Analyse et Conception
- 3 Librairie
- 4 Application**
- 5 Conclusion

## Objectifs de l'architecture conçue

- Avoir une application modulable et facilement extensible
- Fonctionnement identique pour les classes principales en réseau ou en local
- Pouvoir rajouter facilement des outils
- Séparer le traitement du rendu



# Diagramme de classes



L'application est utilisable en local et en réseau, avec un fonctionnement identique.

Les fonctionnalités implémentées sont les suivantes :

- Un outil pour changer la couleur et la forme du pinceau
- Un outil gomme
- Fonctionnalité permettant d'exporter le dessin
- Possibilité de nettoyer complètement le tableau
- Mode plein-écran avec le dessin uniquement (vidéoprojecteur)
- Utilisation simultanée par plusieurs utilisateurs
- Application traduite et packagée (en .deb)

## Fonctionnement en local

- Étalonnage selon la méthode voulue, choix du mode local
- Détection d'un mouvement, dessin directement sur le tableau en respectant les options

L'étalonnage se déroule en plusieurs phases.

- Choix de la webcam et de méthode de suivi
- Choix de l'objet à suivre à partir d'une photo, en l'entourant d'un rectangle
- Réglage du seuil de tolérance à partir du retour de l'étalonnage
- Choix du mode : réseau ou local

# Utilisation de l'application

L'interface permet de visualiser le flux vidéo, et le dessin.  
Les mouvements sont détectés, et le dessin est effectué à partir de ces mouvements.

## Objectifs atteints

- Application fonctionnelle et utilisable
- Beaucoup d'outils voulus implémentés

## Objectifs atteints

- Application fonctionnelle et utilisable
- Beaucoup d'outils voulus implémentés

## Difficultés et ouverture

- Faire une interface gestuelle pour sélectionner gomme, couleur et forme
- Améliorer la gestion du dessin
- Relancer l'étalonnage sans relancer l'application

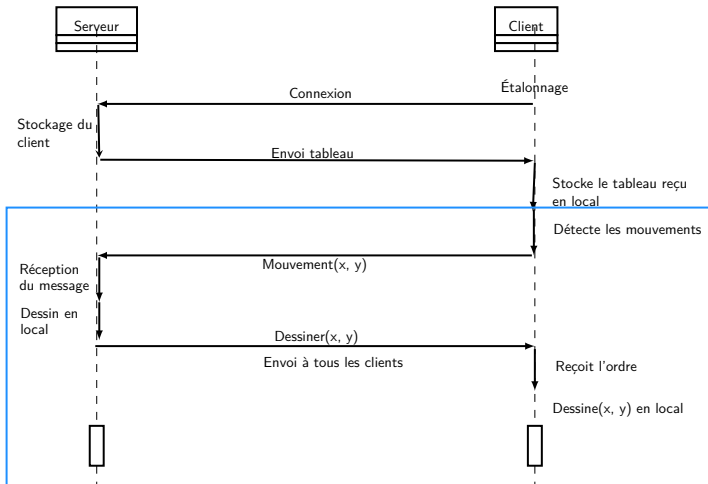
## Fonctionnement en réseau

- Étalonnage selon la méthode voulue, choix du mode réseau
- Récupération du dessin actuel par le client
- Détection d'un mouvement
- Envoi au serveur de ce mouvement (et des options) en respectant le protocole
- Réception du paquet côté serveur, dessin du serveur
- Envoi aux clients de ce point, avec les options (épaisseur, couleur)
- Réception côté client, et dessin en local



# Fonctionnement global : schéma

Déroulement du fonctionnement en réseau de l'application :



Répéter à chaque mouvement détecté

## Objectifs atteints

- Architecture identique au fonctionnement en local
- Application fonctionnelle en mode réseau malgré la difficulté
- Tous les outils fonctionnent

## Objectifs atteints

- Architecture identique au fonctionnement en local
- Application fonctionnelle en mode réseau malgré la difficulté
- Tous les outils fonctionnent

## Difficultés et ouverture

- Optimiser fortement le mode réseau pour réduire les problèmes de lenteur
- Mettre par défaut une couleur à chaque utilisateur

- 1 Introduction
- 2 Analyse et Conception
- 3 Librairie
- 4 Application
- 5 Conclusion**

# Conclusion

## Objectifs atteints

- Solution fonctionnelle
- Respect du cahier des charges
- Découverte (Technologies, gestion de projet...)

# Conclusion

## Objectifs atteints

- Solution fonctionnelle
- Respect du cahier des charges
- Découverte (Technologies, gestion de projet...)

## Difficultés

- Collaboration : Développement incrémental qui oblige à beaucoup communiquer
- Formation : Traitement de l'image, Conception d'architectures
- Techniques : Architecture, Fuites de mémoire...

# Conclusion

## Objectifs atteints

- Solution fonctionnelle
- Respect du cahier des charges
- Découverte (Technologies, gestion de projet...)

## Difficultés

- Collaboration : Développement incrémental qui oblige à beaucoup communiquer
- Formation : Traitement de l'image, Conception d'architectures
- Techniques : Architecture, Fuites de mémoire...

## Ouverture

- Diversifier et optimiser les méthodes de suivi
- Rajouter des fonctionnalités côté application

Merci pour votre attention.