

HAUTE ÉCOLE D'INGÉNIERIE ET DE GESTION DU CANTON DE VAUD

PROJET DE GROUPE (PDG)

RAPPORT DE PROJET

DrawTable

Auteurs :

Sacha BRON

David VILLA

Paul NTAWURUHUNGA

Yassin KAMMOUN

Marc PELLET

Superviseur :

Dr. René RENTSCH

31 décembre 2015



HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch



Table des matières

1	Introduction	11
1.1	Description du projet	11
1.2	Objectif du projet	13
1.3	Technologies utilisées	13
1.4	Équipe de projet	13
1.5	Cadre de réalisation	13
1.6	Choix du sujet	14
2	Conception	15
2.1	Système global	15
2.1.1	Architecture	16
2.1.2	Diagramme d'activités	16
2.1.3	Diagramme de classes	16
2.2	Système de tracking	17
2.2.1	Architecture	17
2.2.2	Diagramme d'activités	17
2.2.3	Diagramme de classes	17
2.3	Système de dessin	18
2.3.1	Architecture	19
2.3.2	Diagramme d'activités	21
2.3.3	Diagramme de classes	21
2.3.4	Gestionnaire de communication réseau	21
2.3.5	Gestionnaire de tracking	21
2.3.6	Contrôleur système	21
2.3.7	Contrôleurs de dessin	22
2.3.8	Dessin	24
2.4	Stylet	25
2.4.1	Modèle à une LED	25
2.4.2	Modèle à deux LEDs	26
2.4.3	Modèle laser	27
2.5	Environnement de dessin	28

2.5.1	Matériel requis	28
2.5.2	Mise en place du matériel	28
2.5.3	Contraintes	28
2.5.4	Exemple d'environnement	28
3	Description technique	31
3.1	Vue d'ensemble des projets	31
3.2	Structure des projets	31
3.3	Patrons de conception	31
3.4	Librairies	31
3.5	Interface graphique utilisateur	31
3.5.1	Structure de l'interface utilisateur	31
3.5.2	Barre des menus	31
3.5.3	Barre d'outils	31
3.5.4	Zone de dessin	31
3.5.5	Ressources	31
3.6	Tracking	31
3.6.1	Système de détection	31
3.6.2	Mouvement du stylet	31
3.6.3	Déclenchement du dessin	31
3.6.4	Filtrage de couleurs	31
3.6.5	Précision du tracking	31
3.6.6	Contraintes de luminosité	31
3.7	Dessin	32
3.7.1	Outils de dessin	32
3.7.2	Outils de formes	32
3.7.3	Outils de couleurs	32
3.7.4	Choix d'épaisseur	32
3.8	Interfaçage tracking-dessin	32
3.8.1	Communication inter-projet	32
3.8.2	Simulation et contrôle de la souris	32
3.9	Historique des actions	32
3.10	Sérialisation	32
3.11	Impression	32
4	Tests & Validation	33
4.1	Stratégie de tests	33
4.2	Outils	33
4.3	Procédures de tests	33
4.4	Résultats	33
5	Problèmes connus	35
6	Conclusion	37

6.1	Solution proposée	37
6.1.1	Fonctionnalités implémentées	37
6.1.2	Fonctionnalités manquantes	37
6.1.3	Propositions d'amélioration	37
6.2	Problèmes rencontrés	37
6.2.1	Problèmes organisationnels	37
6.2.2	Problèmes techniques	37
6.2.3	Problèmes de planification	37
6.3	Respect du planning	37
6.3.1	Planification initiale	37
6.3.2	Évolution	37
6.4	Déroulement du projet	37
6.4.1	Points positifs	37
6.4.2	Points négatifs	37
6.5	Synthèse	37
A	Cahier des charges	41
B	Journal de travail	53
C	Planification	56

Table des figures

1.1	Exemple d'environnement de dessin	12
2.1	Architecture globale du système	16
2.2	Architecture du système de dessin	19
2.3	Hiérarchie des contrôleurs d'outils	22
2.4	Substitution des contrôleurs d'outils	23
2.5	Structure logique d'un dessin	24
2.6	Exemple d'environnement de dessin - vue en perspective	29
2.7	Exemple d'environnement de dessin - vue de gauche	30
A.1	Mockup de l'interface graphique utilisateur	42
A.2	Prototype initial du stylet	44
A.3	Vue en perspective d'un environnement idéal	47
A.4	Vue de côté d'un environnement idéal	48
A.5	Planification initiale des tâches	49
C.1	Planification initiale du projet	57

Liste des tableaux

1.1	Équipe de projet	13
2.1	Mapping outil - contrôleur d'outil	22

Introduction

Ce chapitre se veut être une introduction de ce rapport de travail. Il s'agit dans un premier temps de rappeler le sujet du projet par une description complète de celui-ci. Viennent ensuite l'énumération, le commentaire et l'explication des objectifs que cherche à atteindre ce projet. Les technologies utilisées pour la réalisation du système sont introduites par la suite. Les différents membres constituant l'équipe de projet sont présentés. Cette introduction décrit précisément les rôles de chacun des protagonistes. Finalement, un commentaire quant à la décision de choisir un tel sujet de projet est exposé. Cela consiste à partager les motivations et les raisons qui ont poussé le groupe à partir sur un tel projet.

1.1 Description du projet

Le projet consiste en un outil de dessin tout à fait standard. Celui-ci permet entre autres de dessiner sur un espace de travail. Pour ce faire, l'utilisateur dispose d'un éventail d'outils :

- Outils de dessin : les outils de dessin mettent à disposition un crayon et une gomme permettant de dessiner sans restriction n'importe quelle forme géométrique.
- Outils de formes : les outils de formes mettent à disposition un ensemble de formes géométriques prédéfinies pouvant être dessinées au sein d'un dessin et redimensionnées à la guise de l'utilisateur.
- Outils de couleurs : les outils de couleurs mettent à disposition une palette de couleurs laquelle permet de définir la couleur du trait aussi bien pour les outils de dessin que pour les outils de formes.
- Outils d'épaisseur : les outils d'épaisseur permettent de définir selon une liste prédéfinie l'épaisseur du trait aussi bien pour les outils de dessin que pour les outils de formes.

Malgré ce côté simpliste du système, celui-ci se démarque des outils de dessin traditionnels par le fait que l'espace de travail du dessinateur est non pas l'écran de l'utilisateur mais un support physique tel un mur, un sol, une table ou n'importe quelle autre surface susceptible de jouer le rôle de support de dessin. L'idéal est bien évidemment une surface plane. Toutefois, le système n'est pas restreint par une telle propriété. Celle-ci pourrait tout aussi bien être théoriquement abrupte, instable et bicornue. L'utilisateur définit lui-même ce qu'il juge être un support de dessin propice pour travailler.

Le fait de dessiner sur un support physique plutôt que virtuel nécessite une substitution de la souris de l'ordinateur à un outil plus adéquat pour dessiner. Ceci est rendu possible par la mise à disposition d'un stylet au dessinateur. Ce stylet est conçu sur mesure par l'équipe de projet pour les besoins du système. Il reste toutefois un outil expérimental avec une casquette de prototype. En effet, celui-ci n'a pour but que de valider la conception et l'implémentation du système. Il n'en demeure pas moins que ce stylet reste relativement complexe pour accomplir une telle tâche.

Le stylet se présente sous la forme d'un stylo tout à fait usuel. Toutefois, de par l'objectif de son utilisation, il est caractérisé par les composants suivants :

- Une LED rouge.
- Une LED verte.
- Un interrupteur.
- Une résistance.
- Une batterie.
- Un boîtier.

Bien que l'action de dessiner soit réalisée sur un support physique, le dessin à proprement parlé n'est pas gravé sur ce support. Les faits et gestes du dessinateur avec le stylet sont capturés par une caméra. Un programme informatique reçoit en permanence en provenance de cette caméra des informations liées aux faits et gestes du stylet manipulé par le dessinateur. C'est là qu'intervient le mécanisme du tracking, c'est-à-dire la détection et le suivi du stylet. Toutes ces informations récupérées sont communiquées à un autre programme informatique. Ce dernier stocke, analyse, traite et reproduit ces mêmes faits et gestes de manière à reconstituer logiquement et graphiquement le dessin correspondant.

Afin que le dessinateur puisse disposer d'un retour instantané de son dessin, un projecteur projette la reproduction fidèle du dessin réalisée au sein du second programme informatique. Ainsi, le dessinateur a l'illusion de dessiner directement sur son support physique. Il dispose en plus de cela des fonctionnalités usuelles de sérialisation de dessin. Il peut en effet enregistrer son travail et le reprendre ultérieurement.

La figure suivante illustre un exemple d'environnement de dessin :

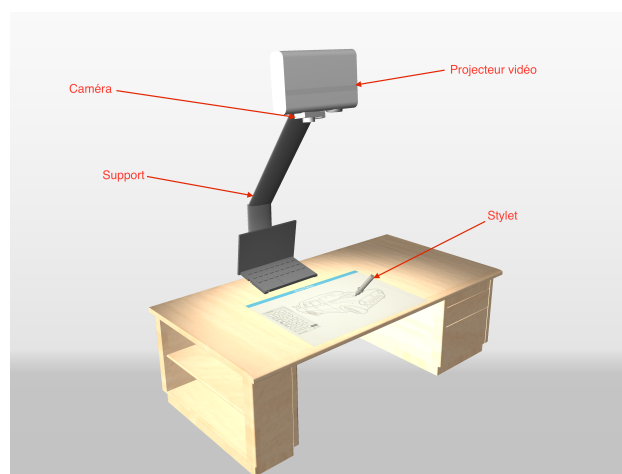


FIGURE 1.1 – Exemple d'environnement de dessin

1.2 Objectif du projet

L'objectif de ce projet est de concevoir un outil de dessin assisté par ordinateur permettant à l'utilisateur de réaliser ses dessins de la manière la plus naturelle possible. À terme, l'utilisateur dessinera directement sur sa table ou n'importe quelle autre surface plane à l'aide d'un stylet et son dessin sera projeté sur son plan de travail, donnant ainsi à l'utilisateur l'impression de dessiner avec un crayon et une feuille.

1.3 Technologies utilisées

Les technologies utilisées pour le développement de l'application et le suivi du stylet sont les suivantes :

- Qt 5.5 : Le framework Qt est une API orientée objet offrant des composants d'interface graphique, d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, Dans le cadre du projet, elle est utilisée pour construire l'interface graphique utilisateur de l'application. Pour de plus amples informations, veuillez-vous référer au site officiel <http://www.qt.io>.
- OpenCV 3.1 : Open Computer Vision (OpenCV) est une bibliothèque graphique libre spécialisée dans le traitement d'images en temps réel. La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités pour le traitement d'images, le traitement vidéos, les calculs matriciels, Dans le cadre du projet, elle est notamment utilisée pour le tracking du stylet. Pour de plus amples informations, veuillez-vous référer au site officiel <http://opencv.org>.
- C++11 : Le C++ est utilisé en guise de langage de programmation de base permettant de manipuler les différentes librairies utilisées dans le cadre du développement du projet. La librairie OpenCV et le framework Qt utilisant par défaut le langage C++, celui-ci constitue donc un choix idéal pour le développement du système.

1.4 Équipe de projet

Le tableau suivant présente l'équipe de projet, la hiérarchie au sein du groupe ainsi que les rôles joués par les différents membres du groupe :

Nom, prénom	E-mail	Hiérarchie	Rôles
Bron, Sacha	sacha.bron@heig-vd.ch	Chef de projet	TODO
Villa, David	david.villa@heig-vd.ch	Chef suppléant	TODO
Kammoun, Yassin	yassin.kammoun@heig-vd.ch	-	TODO
Ntawuruhunga, Paul	paul.ntawuruhunga@heig-vd.ch	-	TODO
Pellet, Marc	marc.pellet@heig-vd.ch	-	TODO

TABLE 1.1 – Équipe de projet

1.5 Cadre de réalisation

Ce projet s'inscrit dans le cadre du cours de Projet de Groupe (PDG) au sein de la Haute École d'ingénierie et de gestion du canton de Vaud (Heig-VD) sis à Yverdon-les-Bains. Selon le plan d'études de l'école, il est dispensé aux étudiants IL du département des Technologies de l'Information et de la Communication (TIC) pour le compte de leur troisième année de formation Bachelor. Le but de ce cours

est d'effectuer un projet en passant par toutes les étapes de développement. Cela inclut le choix d'un sujet, la définition d'un cahier des charges, une phase de recherche et d'analyse suivie du développement de l'application et d'une phase de tests et de validation. En dernière instance, un rapport sur le déroulement du travail et une présentation du projet sont requis dans le but d'évaluer le travail effectué.

1.6 Choix du sujet

Le choix d'un tel sujet se justifie par le fait que ce projet exige un important travail de recherche, de découverte et d'apprentissage de nouvelles technologies comme cela peut notamment être le cas pour la librairie OpenCV. Bien évidemment, ce genre de projet présente des risques compte tenu du fait que la technologie n'est connue de personne, qu'elle peut être difficile à appréhender, à mettre en oeuvre et à maîtriser, que la faisabilité du projet n'est pas facilement définissable et que le temps d'apprentissage est difficilement estimable. Toutefois, toutes ces problématiques ne rendent le projet que plus motivant, attrayant et intéressant.

Ce projet s'inscrivant dans un cadre académique, il s'agit donc d'une opportunité idéale d'acquérir de nouvelles connaissances. Le sujet en lui-même est des plus intéressants. Il se démarque clairement de la monotonie des applications développées dans un tel contexte. Des projets similaires ont certainement déjà été réalisés que ce soit dans un cadre professionnel que dans un cadre académique mais à bien moindre mesure ce qui laisse énormément de place pour la créativité et l'innovation. Par ailleurs, la complexité d'un tel projet ne rend que plus grand le mérite une fois le travail terminé avec un système tout à fait fonctionnel.

D'un point de vue organisationnel, ce projet présente la particularité d'être subdivisé en deux sous-projets étant donné que deux programmes sont développés : l'outil de dessin et le système de tracking du stylet. Un troisième sous-projet pourrait encore ressortir de ces deux derniers puisque la conception et la fabrication du stylet est un travail à part entière. Ainsi, un défi clairement établi de ce projet est l'intégration des différentes composantes pour finalement former un tout.

En dernière instance, ce sujet de projet a été choisi en premier lieu par son originalité puisque il s'avère fort différent des travaux réalisés dans le cadre de la formation Bachelor. Par ailleurs, le fait qu'il présente une difficulté certaine et un risque potentiel d'échec rend le projet d'autant plus motivant à réaliser. Enfin, l'étude d'une nouvelle technologie telle que la librairie OpenCV jusque-là parfaitement inconnue à l'équipe est attrayante puisqu'elle permet à chaque membre du groupe d'enrichir son bagage technique.

Conception

Ce chapitre se veut être une description complète aussi bien de la conception du système que de la conception du stylet. L'architecture débute le descriptif du système par la présentation et la description de ses différentes composantes. Un diagramme de classes concis représentant l'ensemble du système est introduit. Viennent ensuite les présentations des deux composantes logicielles du système : le mécanisme de tracking et l'outil de dessin. Chacune fait l'objet d'une description plus approfondie de son architecture tout comme leur diagramme de classes respectif. Une décomposition de leur structure interne est effectuée de manière à décrire de manière indépendante le rôle et le fonctionnement de chaque constituant du système donné. Le stylet pour sa part est introduit en premier lieu par une description de son principe d'utilisation. Une présentation des différents composants matériels qui le constitue est ensuite réalisée. Un exposé du prototype de stylet est effectué au moyen de schémas. Les étapes de fabrication de celui-ci sont énumérées et commentées en long et large par la suite. Enfin, le fonctionnement du stylet est introduit. En dernière instance, l'environnement de dessin idéal est schématisé à l'aide de modélisations accompagnées de descriptions.

2.1 Système global

Cette section présente le système global du projet. Il s'agit d'exposer, de décrire et de commenter son architecture globale d'une part. D'autre part, il s'agit d'introduire le diagramme de classes correspondant. Celui-ci est toutefois concis dans la mesure où seules les entités sont mentionnées, ni méthode ni attribut n'y figurent dans le but bien évidemment d'alléger le contenu du schéma.

Le système met en jeu différents composants. Certains de ces composants sont de l'ordre du logiciel ; d'autres sont de l'ordre du matériel. Cela se justifie par le fait que ce projet met non seulement en jeu différents composants de nature diverse mais en plus, ces composants sont supposés interagir et ce, malgré leur nature.

La relation entre composants peut prendre différentes formes. Certains composants interagissent entre eux par le biais d'une communication réseau. Certains encore utilisent des périphériques pour rediriger de l'information vers une sortie matérielle donnée. D'autres encore exploitent une entrée matérielle d'autres dispositifs pour récupérer des données.

2.1.1 Architecture

La figure suivante présente l'architecture globale du système. Elle met en évidence entre autres les différentes composantes constituant le système :

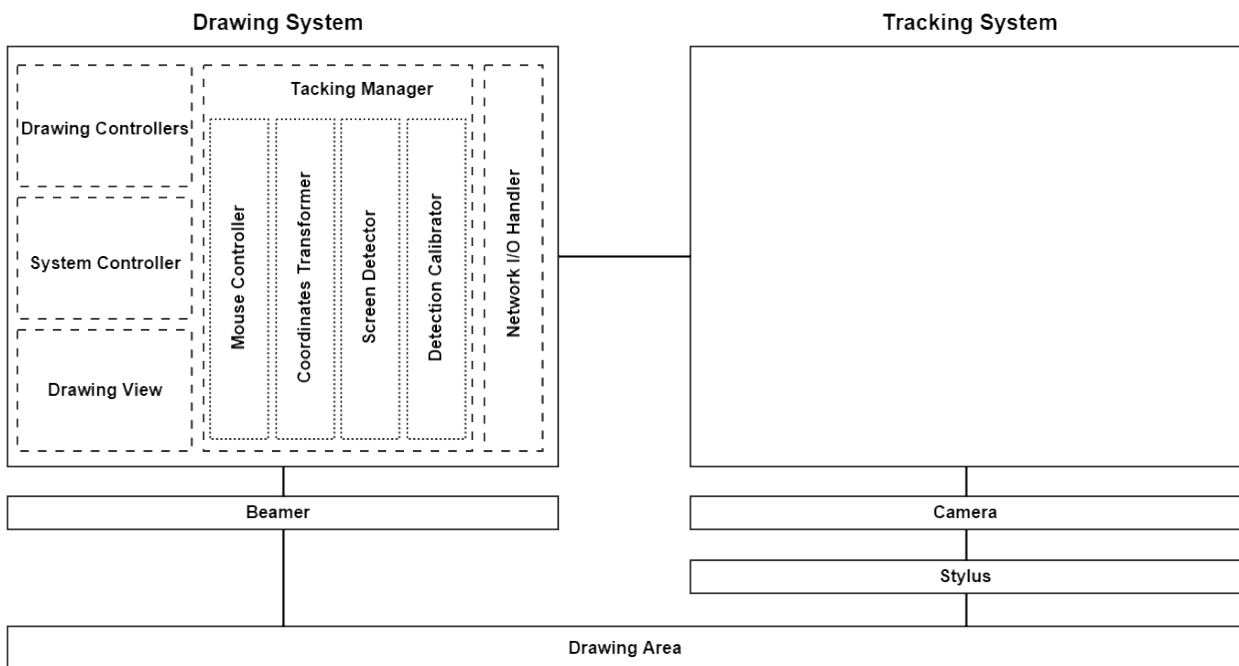


FIGURE 2.1 – Architecture globale du système

TODO : décrire l'architecture globale du système.

2.1.2 Diagramme d'activités

TODO : définir et décrire un diagramme d'activités mettant en jeu les différentes composants du système (Drawing System, Tracking System, Beamer, Drawing Area, Camera, Stylus).

2.1.3 Diagramme de classes

TODO : définir et décrire le diagramme de classes de l'ensemble du système (avec entités, sans méthodes, sans attributs, sans cardinalités).

2.2 Système de tracking

TODO : introduire le contenu de la section (qu'est-ce qui y sera présenté dans l'ordre).

2.2.1 Architecture

TODO : décrire l'architecture du système de tracking.

2.2.2 Diagramme d'activités

TODO : définir et décrire un diagramme d'activités mettant en jeu le système de tracking (Tracking System), les constituants de sa structure interne, le système de dessin (Drawing System), la caméra (Camera), le stylet (Stylus) et le support de dessin (Drawing Area).

2.2.3 Diagramme de classes

TODO : définir et décrire le diagramme de classes complet du système de tracking (avec entités, avec attributs, avec méthodes, avec cardinalités).

2.3 Système de dessin

Cette section présente le système de dessin. Il s'agit d'exposer, de décrire et de commenter son architecture d'une part. D'autre part, il s'agit d'introduire le diagramme de classes correspondant. Contrairement à celui du système global, le diagramme de classes ici est complet : attributs, méthodes, cardinalités et toute autre information jugée pertinente y figurent. Enfin, la structure interne du système y est décomposée. Chacune de ses constituantes y est décrite de manière indépendante.

Le système de dessin correspond à un outil logiciel lequel fournit une interface graphique utilisateur destinée à y produire virtuellement du dessin. Celui-ci a pour but de représenter logiquement et graphiquement sur un moniteur le dessin qu'un utilisateur est en train de dessiner sur son support physique qui, pour rappel, est supposé être n'importe quelle surface plane.

Cet outil de dessin a également pour objectif de fournir à l'utilisateur un ensemble de fonctionnalités lui permettant de manipuler son dessin. Ainsi, il se voit offrir un éventail d'outils :

- Outils de dessin :
 - crayon, gomme.
- Outils de formes :
 - cercle, rectangle.
- Outils de couleurs :
 - palette de couleurs.
- Outils d'épaisseur :
 - épaisseur de trait.

Outre les fonctionnalités de dessin, l'outil se doit de proposer les fonctionnalités usuelles d'un logiciel dans la mesure où il s'agit d'une application manipulant un document, c'est-à-dire une unité d'informations lisible et susceptible d'être stockée de manière persistante. Dans le cas présent, il s'agit d'un dessin. Ainsi, les fonctions d'importation et de sauvegarde, d'impression et d'historique d'actions se doivent d'être disponibles, donc conçues et implémentées.

Selon ce qui a été défini dans le cahier des charges, les fonctionnalités suivantes sont donc mises à disposition dans l'application de dessin :

- Sérialisation :
 - enregistrement, importation, impression.
- Historique d'actions :
 - annulation, rétablissement.

Compte tenu du fait que le système de dessin est susceptible de communiquer avec d'autres composantes du système global, notamment des composantes matérielles, des fonctionnalités supplémentaires se révèlent être nécessaires. Il s'agit notamment de permettre à l'utilisateur de pouvoir configurer les composantes matérielles interagissant avec les composantes logicielles de manière à ce qu'elles puissent fonctionner et communiquer de manière optimale avec les meilleurs paramètres possibles et donc, avec le meilleur taux de performance.

2.3.1 Architecture

La figure suivante présente l'architecture du système de dessin :

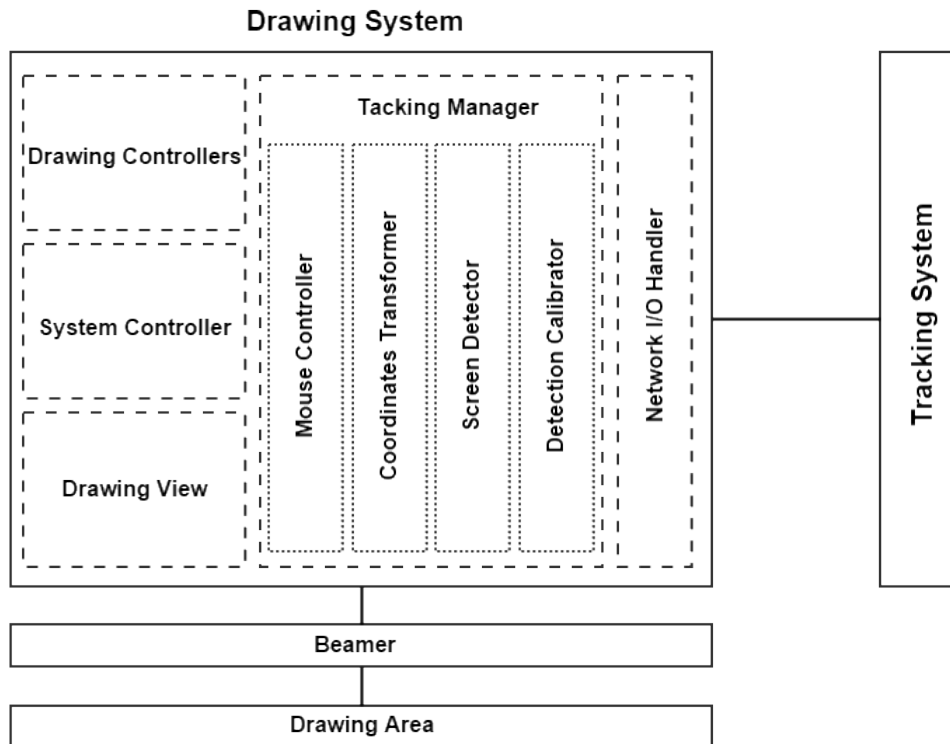


FIGURE 2.2 – Architecture du système de dessin

Le système de dessin est composé d'un certain nombre de constituants, chacun réalisant des tâches bien déterminées. Par ailleurs, le système de dessin fait l'objet d'une dépendance avec le système de tracking. En effet, ceux-ci interagissent ensemble par le biais d'une communication réseau.

La communication réseau avec le système de tracking est gérée du point de vue du système de dessin par un gestionnaire de communication réseau, symbolisé par le bloc *Network I/O Handler* sur la figure 2.2. Ce gestionnaire se comporte à la manière d'un *listener*, d'un écouteur d'événements qui dans le cas présent consistent en la réception de messages. Ainsi, il se voit écouter tous les messages émis par le composant réseau du système de tracking. Les informations qui lui sont communiquées consistent typiquement en la description abstraite des faits et gestes de l'utilisateur avec le stylet sur le support physique, c'est-à-dire le bloc *Drawing Area*. Cette description correspond typiquement en des actions survenues à des coordonnées données. Lors de la réception de tels messages, le gestionnaire de communication réseau transmet ces informations au gestionnaire de tracking du système de dessin, symbolisé par le bloc *Tracking Manager* sur la figure 2.2.

Le bloc *Tracking Manager* remplit plusieurs rôles. Concrètement, il accomplit les tâches suivantes :

- Simuler la souris : il s'agit de simuler et de reproduire avec la souris de l'ordinateur les faits et gestes de l'utilisateur avec le stylet. Cela est assuré par le bloc *Mouse Controller*.
- Transformer des coordonnées : il s'agit de transformer les coordonnées du point de vue de la caméra vers le point de vue de l'écran. Cela est assuré par le bloc *Coordinates Transformer*.

- Détecter la projection d'écran : il s'agit de détecter la zone physique sur laquelle la projection de la reproduction du dessin est réalisée. Cela est réalisé par le biais du bloc *Screen Detector*.
- Calibrer la détection : il s'agit de calibrer la détection du stylet sur le support physique, c'est-à-dire de le localiser de manière précise en vue de l'exécution du mécanisme de tracking du stylet. Cette calibration est effectuée par le bloc *Detection Calibrator*.

En définitive, le gestionnaire du tracking veille d'une part à ce que les composantes matérielles du système soient disponibles et fonctionnent de la manière la plus optimale possible. D'autre part, ce même gestionnaire tente tant bien que mal de reproduire ce que cherche à dessiner l'utilisateur en simulant ces faits et gestes et en les reportant sur l'équivalent logique du support physique de dessin.

L'ensemble des constituants du système de dessin est géré par un contrôleur général, ou contrôleur système, symbolisé par le bloc *System Controller* sur la figure 2.2. Celui-ci fait office de médiateur entre les différents constituants et veille à transmettre des messages lorsque besoin se fait sentir. Concrètement, il délègue à un constituant donné une tâche émise par un autre constituant. Ce contrôleur général est donc le noyau central du système de dessin, le coeur du système dit autrement.

Différents outils sont mis à la disposition de l'utilisateur en vue de dessiner et de personnaliser ses esquisses. Pour ce faire, différents contrôleurs sont définis de manière à traiter les interactions de l'utilisateur avec un outil donné. En conséquence, il y a autant de contrôleurs de dessin que d'outils prévus pour dessiner. Ces contrôleurs de dessin, ou contrôleurs d'outils de dessin, sont représentés par le bloc *Drawing Controllers*. Ceux-ci sont mis à disposition du contrôleur général, *System Controller* autrement dit.

Lorsque le contrôleur de souris, c'est-à-dire le bloc *Mouse Controller*, simule une action utilisateur, celle-ci est émise au contrôleur général qui délègue la reproduction du geste de l'utilisateur selon l'outil utilisé au contrôleur de dessin approprié. Cette délégation est réalisée de manière abstraite. Le contrôleur général ne sait en aucun cas quel contrôleur de dessin se voit confier la tâche d'appliquer le geste utilisateur sur l'équivalent logique du dessin physique.

La reproduction des faits et gestes de l'utilisateur à l'aide du simulateur de souris, du contrôleur général et d'un contrôleur de dessin adéquat nécessite de disposer d'une vue sur laquelle ce traitement peut avoir lieu. Ceci est rendu possible par la définition d'une vue de dessin, symbolisée par le bloc *Drawing View* sur la figure 2.2. Le dessin que cherchera à dessiner l'utilisateur sera logiquement stocké dans ce constituant, affiché graphiquement et donc visible sur le moniteur de l'utilisateur. C'est précisément un des contrôleurs du bloc *Drawing Controllers* qui effectuera cette tâche à la demande du contrôleur général, *System Controller*, qui lui, aura réagi initialement à la demande du simulateur de souris, *Mouse Controller*, qui aura reçu antérieurement une instruction de la part du système de tracking.

Le bloc *Beamer* est un composant matériel à part entière. C'est vers celui-ci qu'est projetée la reproduction du dessin de l'utilisateur. D'un point de vue global de l'ensemble de l'infrastructure, le système n'accomplit aucune tâche pour que cette projection soit possiblement effective. Ce composant fait bel et bien partie du système dans la mesure où il reçoit indirectement de l'information en provenance d'un composant logiciel, le système de dessin en l'occurrence. Toutefois, cette transmission d'information

n'est pas réalisée par le système présenté mais par le système d'exploitation de la machine hôte. Ainsi, il incombe à l'utilisateur de veiller à ce qu'un tel périphérique soit connecté à la machine hôte et qu'il soit opérationnel. En dernière instance, pour que la projection soit visible, il est attendu de l'utilisateur qu'il duplique ou qu'il étende son écran sur la sortie du projecteur de manière à ce que ce dernier affiche le contenu du moniteur.

Du point de vue du système de dessin, le bloc *Drawing Area* n'est rien de plus qu'une zone sur laquelle est projetée la reproduction du dessin de l'utilisateur par le bloc *Beamer*. Il s'agit tout simplement du support physique de l'utilisateur lequel remplace le support papier traditionnel.

2.3.2 Diagramme d'activités

2.3.3 Diagramme de classes

2.3.4 Gestionnaire de communication réseau

2.3.5 Gestionnaire de tracking

2.3.5.1 Contrôleur de souris

2.3.5.2 Transformateur de coordonnées

2.3.5.3 Détecteur d'écran

2.3.5.4 Mécanisme de calibration

2.3.6 Contrôleur système

2.3.7 Contrôleurs de dessin

L'utilisation d'un outil de dessin nécessite le recours à un contrôleur adéquat devant permettre d'appliquer l'action que représente l'outil en question. Ainsi, dans le cas où l'utilisateur utiliserait la gomme, le contrôleur gérant la gomme devrait reproduire à l'exactitude les faits et gestes de l'utilisateur avec le stylet qui, dans le cas présent, représenterait une gomme.

2.3.7.1 Hiérarchie des contrôleurs

Les contrôleurs d'outils se présentent en plusieurs exemplaires. Chaque contrôleur a pour but de traiter les interactions de l'utilisateur avec son dessin pour un outil donné. Ainsi, si l'utilisateur manipule la gomme, le contrôleur relatif à la gomme entrera en scène et réagira aux actions de l'utilisateur, c'est-à-dire d'effacer là où l'utilisateur le demande. En l'occurrence, il s'agit du contrôleur nommé *EraserController*. Si l'utilisateur manipule le crayon, le contrôleur relatif au crayon entrera en scène et ainsi de suite. Pour ce faire, une hiérarchie de contrôleurs d'outils a été établie. Elle est décrite par la figure suivante :

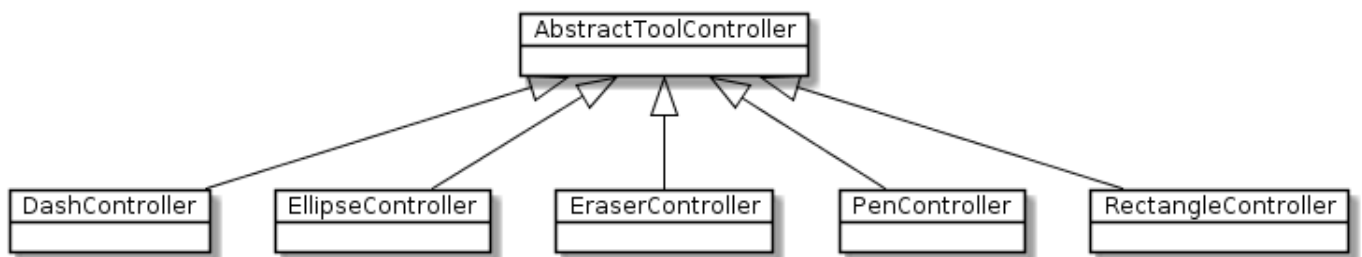


FIGURE 2.3 – Hiérarchie des contrôleurs d'outils

Un contrôleur d'outil abstrait est défini au sommet de la hiérarchie. Celui-ci apporte une couche d'abstraction à l'architecture dans la mesure où le contrôleur général ne sait à aucun moment avec quel contrôleur d'outil il interagit. Ainsi, lorsqu'un contrôleur est soumis à une tâche, le contrôleur général lui déléguera le travail. Par polymorphisme, la bonne implémentation du contrôleur d'outil sera appelée.

À noter qu'il y a autant de spécialisations de contrôleur d'outil qu'il y a d'outils mis à disposition de l'utilisateur. Pour rappel, les outils se présentent en quatre catégories à savoir les outils de dessin, les outils de formes, les outils de couleurs et les outils d'épaisseur. Toutefois, seules les deux premières font l'objet de contrôleurs d'outils, la raison étant due au fait que les outils appartenant à celles-ci sont les seuls permettant de produire du dessin à proprement parlé. Les autres font plutôt offices de cosmétiques pouvant être appliquées au dessin. Compte tenu de la hiérarchie établie et selon ce qui a été dit plus tôt, le mapping établi entre outils et contrôleurs d'outils se présente comme suit :

Outil	Contrôleur d'outil
Crayon	PenController
Gomme	EraserController
Ligne	DashController
Cercle	EllipseController
Rectangle	RectangleController

TABLE 2.1 – Mapping outil - contrôleur d'outil

Bien que non représenté au sein de la figure précédente, les différents contrôleurs d'outil sont régis par les mêmes comportements. Ceux-ci sont effectivement appelés à réagir aux événements suivants :

- Mouvement de la souris.
- Pression du bouton de la souris.
- Relâchement du bouton de la souris.
- Double-click de la souris.

Lorsque le simulateur de souris invoque le contrôleur général suite à la réception d'un message en provenance du système de tracking, le contrôleur général veille à déléguer le travail d'application de dessin au contrôleur d'outil de dessin adéquat en déclenchant l'événement propice. Ainsi, si l'utilisateur sélectionne dans un premier temps l'outil crayon et que dans un second temps, il produit une esquisse avec ce même outil, l'événement de pression du bouton de la souris sera déclenché après interprétation. Le contrôleur général déclenchera ce même événement pour invoquer le contrôleur *PenController* qui tâchera de reproduire l'esquisse de l'utilisateur.

2.3.7.2 Substitution des contrôleurs

Selon l'outil sélectionné par l'utilisateur, les contrôleurs substituent les uns autres selon le courant. Ce mécanisme est géré d'une part par l'interface graphique utilisateur. Pour rappel, celle-ci est représentée par la classe *MainWindow* pour laquelle des événements sont susceptibles d'être déclenchés par l'utilisateur comme le choix d'un outil par exemple. Ainsi, un tel déclenchement doit avoir pour conséquence la substitution des contrôleurs. La sélection d'un outil impliquera que le contrôleur général change le contrôleur d'outil courant sans pour autant savoir lequel il s'agit. Le diagramme de séquence qui suit récapitule le comportement attendu :

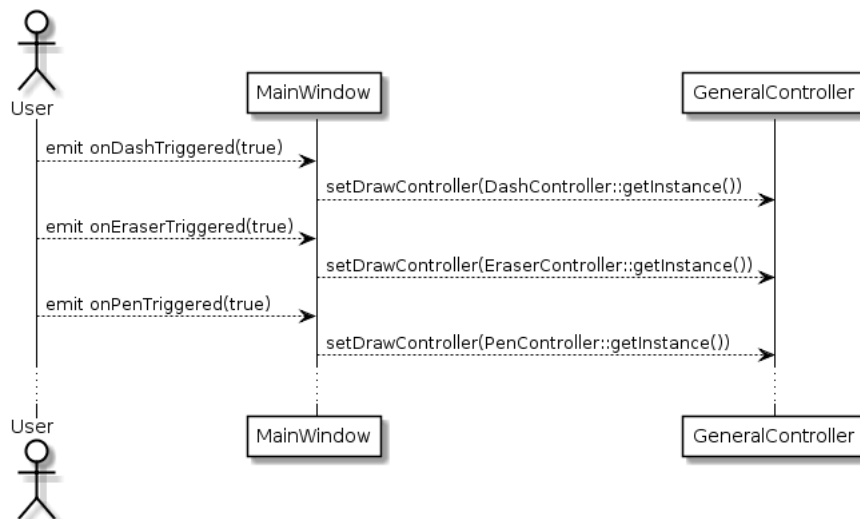


FIGURE 2.4 – Substitution des contrôleurs d'outils

La figure précédente résume parfaitement ce qui a été décrit plus tôt ; l'utilisateur sélectionne un outil. Cette action a pour conséquence le déclenchement d'un événement adéquat au sein de l'interface graphique utilisateur. Celle-ci fait ensuite appel au contrôleur général pour mettre à jour le contrôleur d'outil courant.

2.3.8 Dessin

Le concept de dessin correspond à n'importe quelle esquisse réalisée par l'utilisateur au moyen du stylet prévu et conçu à cet effet. Cela peut aller du simple point au milieu d'une surface à la modélisation en perspective du bâtiment du site de Cheseaux de la Haute École d'ingénierie et de gestion du canton de Vaud (Heig-VD).

2.3.8.1 Vue virtuelle

Bien qu'il soit prévu que le dessin esquissé soit projeté sur son support physique, le dessin logique du dessin doit être représenté sur le moniteur de l'utilisateur. En conséquence, une vue virtuelle du dessin doit être mise à disposition du système. Celui-ci fait entre autres appel aux contrôleurs de dessin pour y stocker logiquement et y retranscrire graphiquement les esquisses de l'utilisateur à la demande du contrôleur général.

2.3.8.2 Structure logique

La structure logique du dessin se base sur une structure de type *bitmap*, c'est-à-dire une matrice où chaque élément de la matrice correspond à un pixel. La valeur stockée dans un tel élément consiste en la couleur de ce pixel. Cette couleur est représentée quant à elle par un code hexadécimal. En guise d'exemple, la figure suivante illustre une telle structure.

```
0000FF 0000FF 0000FF 0000FF 0000FF 0000FF 0000FF 0000FF
00FF00 FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF 00FF00
00FF00 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 00FF00
00FF00 FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF 00FF00
00FF00 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 00FF00
00FF00 FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF 00FF00
00FF00 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 00FF00
0000FF 0000FF 0000FF 0000FF 0000FF 0000FF 0000FF 0000FF
```

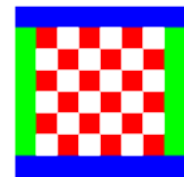


FIGURE 2.5 – Structure logique d'un dessin

2.3.8.3 Représentation graphique

La structure logique d'un dessin doit être traduite de manière à pouvoir obtenir une représentation graphique de celui-ci. Cette représentation graphique correspond au dessin affiché sur l'interface graphique utilisateur de l'application, sur le moniteur de l'utilisateur autrement dit.

2.3.8.4 Projection

La projection du dessin doit être réalisée par le biais d'un dispositif matériel adéquat. Dans le cadre de ce projet, il est préconisé de recourir à un projecteur lequel est censé être connecté à l'ordinateur. Le système d'exploitation de son côté veille à rediriger la représentation graphique du dessin vers la sortie matérielle faisant l'interface avec le périphérique de projection.

2.4 Stylet

Cette section décrit la conception du stylet devant remplacer la souris de l'ordinateur pour dessiner sur un support physique. Sa conception ayant évolué en cours de projet, les différentes variantes y sont présentées. Chacune suit la même procédure de description. Les différents composants constituant ce stylet sont décrits de manière détaillée. Cela permet à tout un chacun de pouvoir reproduire cet outil en suivant les mêmes étapes de fabrication avec les mêmes composants voire des composants équivalents. Le prototype issu de la fabrication de l'outil de dessin est par la suite exposé. Finalement, le principe d'utilisation du stylet est présenté et commenté ce qui constitue un guide d'utilisation de celui-ci par la même occasion. Certains choix de conception y sont d'ailleurs justifiés.

2.4.1 Modèle à une LED

Le modèle à une LED du stylet a été défini lors de la définition du cahier des charges.

2.4.1.1 Composants

Le premier modèle du stylet est caractérisé par les composants suivants :

- Une LED rouge.
- Un interrupteur.
- Une résistance.
- Une batterie.
- Un boîtier.

2.4.1.2 Fabrication

TODO : décrire les étapes de fabrication du stylet (celles-ci devraient permettre à tout un chacun de pouvoir le créer à partir de zéro).

2.4.1.3 Prototype

La figure suivante propose une ébauche de prototype du modèle à une LED du stylet :

TODO : inclure un schéma du prototype de stylet à une LED.

2.4.1.4 Fonctionnement

TODO : décrire le fonctionnement interne du stylet.

2.4.1.5 Principe d'utilisation

TODO : décrire le principe d'utilisation du stylet, c'est-à-dire ce qu'il faut faire exactement pour pouvoir dessiner (position du bras, disposition de la main et des doigts, appui sur l'interrupteur).

2.4.2 Modèle à deux LEDs

Le modèle à deux LED du stylet a été défini durant la première partie du projet. Il remplace le modèle à une LED.

TODO : expliquer et justifier la raison du changement de modèle.

2.4.2.1 Composants

Le deuxième modèle du stylet est caractérisé par les composants suivants :

- Une LED rouge.
- Une LED verte.
- Un interrupteur.
- Une résistance.
- Une batterie.
- Un boîtier.

2.4.2.2 Fabrication

TODO : décrire les étapes de fabrication du stylet (celles-ci devraient permettre à tout un chacun de pouvoir le créer à partir de zéro).

2.4.2.3 Prototype

La figure suivante propose une ébauche de prototype du modèle à deux LEDs du stylet :

TODO : inclure un schéma du prototype de stylet à deux LEDs.

2.4.2.4 Fonctionnement

TODO : décrire le fonctionnement interne du stylet.

2.4.2.5 Principe d'utilisation

TODO : décrire le principe d'utilisation du stylet, c'est-à-dire ce qu'il faut faire exactement pour pouvoir dessiner (position du bras, disposition de la main et des doigts, appui sur l'interrupteur).

2.4.3 Modèle laser

Le modèle laser du stylet a été défini durant la deuxième partie du projet. Il remplace le modèle à deux LEDs.

TODO : expliquer et justifier la raison du changement de modèle.

2.4.3.1 Composants

Le troisième modèle du stylet est caractérisé par les composants suivants :

- Une LED rouge.
- Un interrupteur.
- Une résistance.
- Une batterie.
- Un boîtier.

2.4.3.2 Fabrication

TODO : décrire les étapes de fabrication du stylet (celles-ci devraient permettre à tout un chacun de pouvoir le créer à partir de zéro).

2.4.3.3 Prototype

La figure suivante propose une ébauche de prototype du modèle laser du stylet :

TODO : inclure un schéma du prototype de stylet laser.

2.4.3.4 Fonctionnement

TODO : décrire le fonctionnement interne du stylet.

2.4.3.5 Principe d'utilisation

TODO : décrire le principe d'utilisation du stylet, c'est-à-dire ce qu'il faut faire exactement pour pouvoir dessiner (position du bras, disposition de la main et des doigts, appui sur l'interrupteur).

2.5 Environnement de dessin

Cette section présente la conception de l'environnement de dessin. Ce travail de conception est nécessaire dans la mesure où le système ne peut être opérationnel et fonctionnel qu'à la seule condition de disposer d'un environnement de travail adéquat répondant à un certain nombre de pré-requis. Concrètement, il s'agit d'introduire le matériel nécessaire à sa mise en place, de décrire précisément celle-ci et de présenter un exemple d'environnement jugé idéal pour l'utilisation de l'application.

2.5.1 Matériel requis

Les points suivants constituent le matériel requis pour l'utilisation de l'application :

- Un prototype de stylet faisant office d'outil de dessin.
- Une caméra permettant de traquer les mouvements du stylet.
- Un projecteur vidéo permettant de retranscrire le dessin de l'utilisateur.
- Un plan de travail permettant de dessiner.
- Un support prévu pour la disposition de la caméra et du projecteur au-dessus du plan de travail.

Plusieurs remarques s'imposent quant à cette énumération du matériel requis. En premier lieu, la caméra mentionnée correspond à n'importe quel dispositif permettant de capturer des informations vidéos ; il peut donc s'agir aussi bien d'une webcam, d'une micro-caméra d'un smartphone, d'une caméra intégrée à un ordinateur ou même d'une caméra professionnelle. La différence dans le choix de ce matériel influera toutefois la qualité du système de tracking du stylet. Par ailleurs, le projecteur vidéo n'est pas réellement obligatoire dans la mesure où l'utilisateur peut tout à fait dessiner sur un support donné. Il peut tout aussi bien constater le résultat de ses esquisses sur le moniteur de son ordinateur.

2.5.2 Mise en place du matériel

La mise en place du matériel requis est décrite par la procédure suivante :

1. Disposez le support de travail.
2. Posez le stylet sur le support de travail.
3. Placez la caméra au-dessus du support de travail.
4. Placez le projecteur vidéo au-dessus du support de travail.

2.5.3 Contraintes

La procédure de mise en place n'est présentée ici qu'en guise de suggestion d'une disposition possible du matériel requis. Le but n'est pas d'imposer des contraintes matérielles et environnementales à l'utilisateur. Aussi, cette procédure peut effectivement être adaptée selon les besoins de l'utilisateur et/ou selon les possibilités que lui offre son environnement. L'important est de veiller à ce que le stylet demeure visible autant que possible dans le champ de vision de la caméra de manière à pouvoir le détecter et suivre ses mouvements de la manière la plus précise qui soit.

2.5.4 Exemple d'environnement

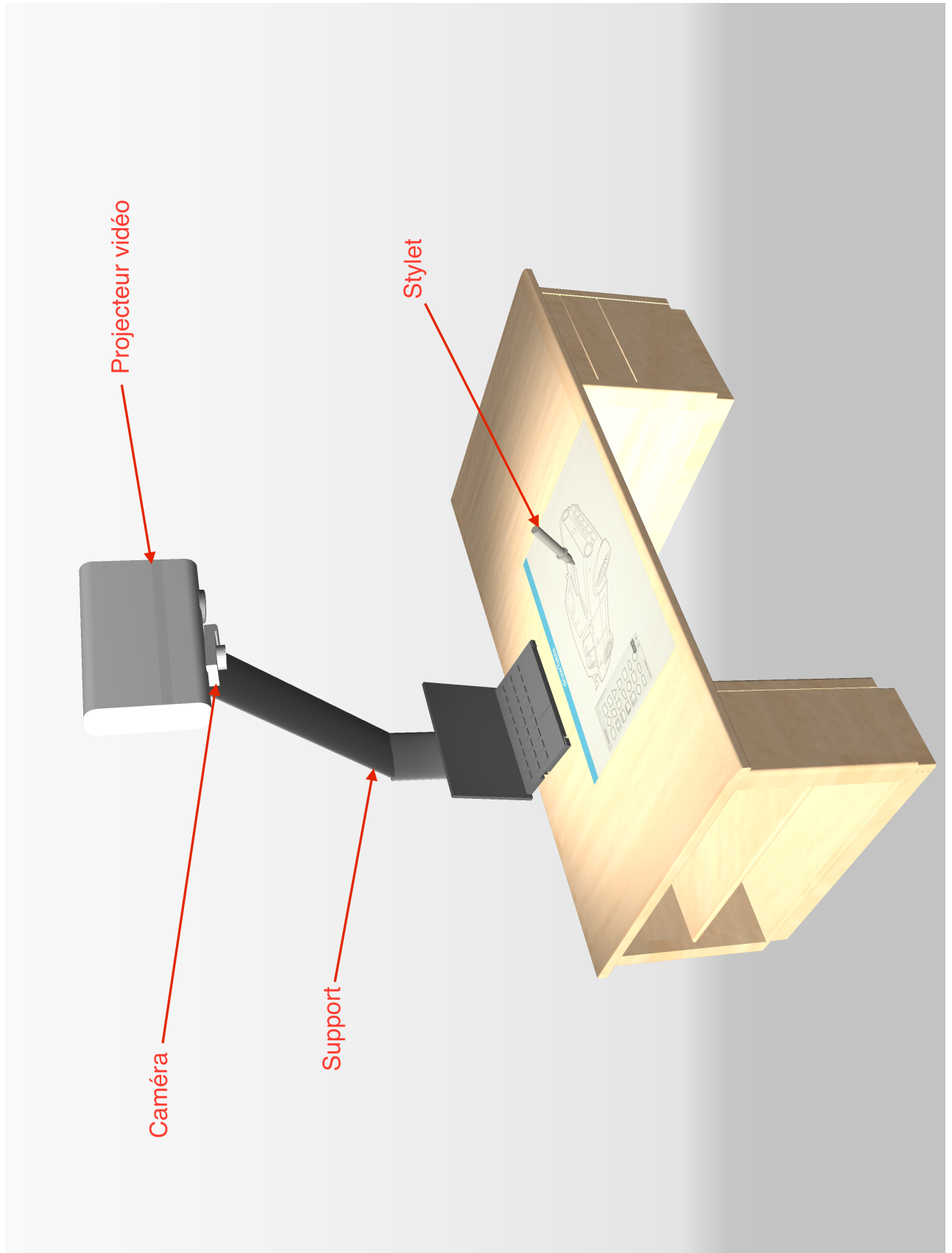


FIGURE 2.6 – Exemple d’environnement de dessin - vue en perspective

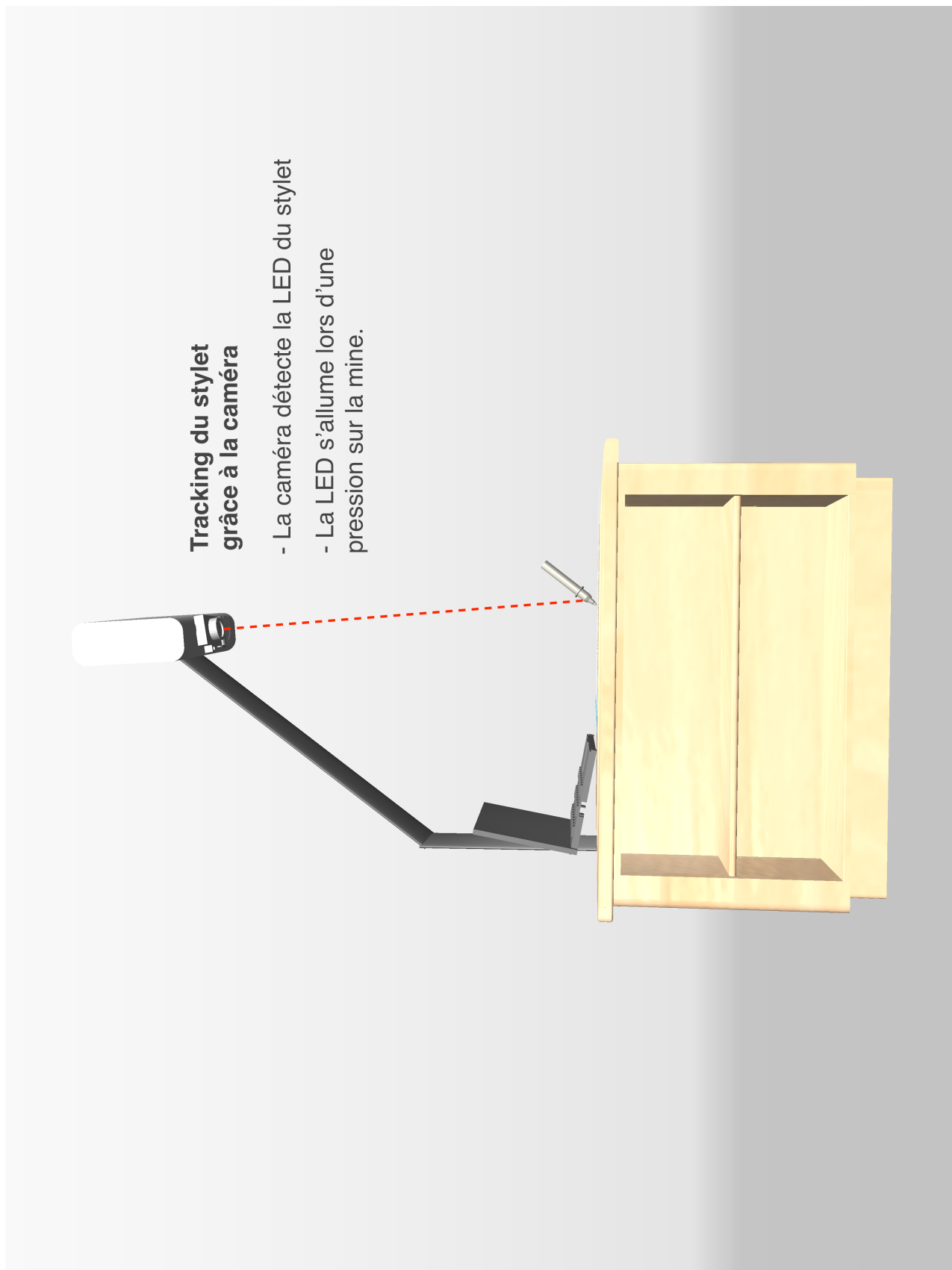


FIGURE 2.7 – Exemple d’environnement de dessin - vue de gauche

Description technique

3.1 Vue d'ensemble des projets

3.2 Structure des projets

3.3 Patrons de conception

3.4 Librairies

3.5 Interface graphique utilisateur

3.5.1 Structure de l'interface utilisateur

3.5.2 Barre des menus

3.5.3 Barre d'outils

3.5.4 Zone de dessin

3.5.5 Ressources

3.6 Tracking

3.6.1 Système de détection

3.6.2 Mouvement du stylet

3.6.3 Déclenchement du dessin

3.6.4 Filtrage de couleurs

3.6.5 Précision du tracking

3.6.6 Contraintes de luminosité

3.7 Dessin

3.7.1 Outils de dessin

3.7.1.1 Crayon

3.7.1.2 Gomme

3.7.2 Outils de formes

3.7.2.1 Trait

3.7.2.2 Rectangle

3.7.2.3 Cercle

3.7.3 Outils de couleurs

3.7.3.1 Palette de couleurs

3.7.4 Choix d'épaisseur

3.8 Interfaçage tracking-dessin

3.8.1 Communication inter-projet

3.8.1.1 Modèle des sockets

3.8.1.2 Protocole de communication

3.8.1.3 Structure des messages

3.8.2 Simulation et contrôle de la souris

3.9 Historique des actions

3.10 Sérialisation

3.11 Impression

Tests & Validation

4.1 Stratégie de tests

4.2 Outils

4.3 Procédures de tests

4.4 Résultats

Chapitre 5

Problèmes connus

Conclusion

6.1 Solution proposée

6.1.1 Fonctionnalités implémentées

6.1.2 Fonctionnalités manquantes

6.1.3 Propositions d'amélioration

6.2 Problèmes rencontrés

6.2.1 Problèmes organisationnels

6.2.2 Problèmes techniques

6.2.3 Problèmes de planification

6.3 Respect du planning

6.3.1 Planification initiale

6.3.2 Évolution

6.4 Déroulement du projet

6.4.1 Points positifs

6.4.2 Points négatifs

6.5 Synthèse

Listings

Cahier des charges

A.1 Contexte

Ce projet de groupe se déroule dans le cadre de la formation Bachelor HES de la Haute-Ecole d'Ingénierie et de Gestion du canton de Vaud. Il compte pour une unité d'enseignement du plan d'étude de l'orientation Informatique logiciel (IL) du département des Technologies de l'Information et de la Communication (TIC) de l'école. Il se réalise durant le cinquième semestre de l'année de formation 2015/16.

A.2 Membres du groupe

Les personnes suivantes constituent les membres du groupe de l'équipe de projet :

- Sacha Bron - *Chef de groupe*
- David Villa - *Chef suppléant*
- Paul Ntawuruhunga
- Yassin Kammoun
- Marc Pellet

A.3 Objectif du projet

L'objectif de ce projet est de concevoir un outil de dessin assisté par ordinateur permettant à l'utilisateur de réaliser ses dessins de la manière la plus naturelle possible. À terme, l'utilisateur dessinera directement sur sa table ou n'importe quelle autre surface plane à l'aide d'un stylet et son dessin sera projeté sur son plan de travail, donnant à l'utilisateur l'impression de dessiner avec un crayon et une feuille.

A.4 Fonctionnalités principales

Les fonctionnalités principales du projet sont les suivantes :

- Dessin à l'aide d'un stylet.
- Projection de l'image sur le plan de travail.

- Alignement de l'image avec la position du stylet.
- Calibrage du stylet.
- Plan de travail avec barre d'outils.
 - Une barre d'outils sera projetée dans une zone du plan travail permettant à l'utilisateur de sélectionner un outil.
 - Outils de dessin : crayon, gomme.
 - Outils de forme : ligne droite, carré, cercle.
 - Outils de couleur : palette de couleurs.
 - Outils de choix d'épaisseur.
- Annulation d'une action
 - Une pile d'actions est enregistrée et un bouton dans la barre d'outils permet de remonter cette pile, annulant ainsi les dernières modifications apportées au dessin.
- Enregistrement et importation du dessin
 - Possibilité d'enregistrer ou importer son dessin au format PNG.

A.5 Fonctionnalités supplémentaires

Les fonctionnalités supplémentaires du projet sont les suivantes :

- Site vitrine.
- Pipette.
- Remplissage.
- Rétablissement d'une action.

A.6 Mockup

La figure suivante illustre une ébauche de l'interface graphique utilisateur devant correspondre au plan de travail :

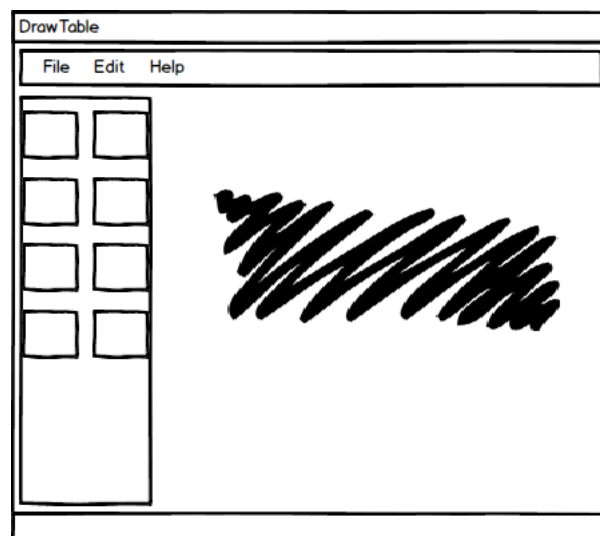


FIGURE A.1 – Mockup de l'interface graphique utilisateur

L'interface graphique utilisateur est divisée en trois parties à savoir la barre des menus, la barre d'outils et la zone de dessin.

A.7 Prototype du stylet

Le prototype de stylet est conçu par le chef de projet. Sa conception nécessite les composants suivants :

- LED.
- Un interrupteur.
- Une résistance.
- Une batterie.
- Un boîtier.

La figure suivante propose une ébauche de prototype du stylet :



FIGURE A.2 – Prototype initial du stylet

A.8 Principe de fonctionnement

Ce projet contient trois principales difficultés : la détection de contact du stylet sur la table, le suivi de la position du stylet (*tracking*) et la précision de l'entier du système.

Pour cela nous avons imaginé un stylet muni d'une mine spéciale. Lors d'une pression sur le plan de travail, cette dernière presse sur un bouton qui enclenche une diode électroluminescente placée proche de la mine. Une caméra placée au-dessus du plan de travail filme la scène et envoie l'image au logiciel chargé de suivre la position de la mine du stylet en détectant cette LED.

Le programme de dessin récupère les coordonnées du stylet et les interprète de manière à pouvoir projeter le dessin sur le plan de travail à l'aide du projecteur vidéo.

L'application comportera différents threads : un thread de l'application sera consacré au tracking de la LED, un autre récupérera les coordonnées pour les interpréter sur le plan de travail en fonction de l'outil sélectionné.

A.9 Public-cible

L'application vise avant toute chose tout amateur de dessin souhaitant se passer d'une feuille.

A.10 Technologies utilisées

Les technologies utilisées pour le développement de l'application et le suivi du stylet sont les suivantes :

- Librairie OpenCV - Pour le suivi du stylet en temps réel.
- Framework Qt - Pour l'interface graphique du plan de travail.

A.11 Matériel requis

Les points suivants constituent le matériel requis pour l'utilisation de l'application.

- Un prototype de stylet faisant office d'outil de dessin.
- Une caméra permettant de traquer les mouvements du stylet.
- Un projecteur vidéo permettant de retranscrire le dessin de l'utilisateur.
- Un plan de travail permettant de dessiner.
- Un support prévu pour la disposition de la caméra et du projecteur au-dessus du plan de travail.

A.12 Plateformes supportées

De par la nature du framework Qt, l'application est automatiquement multiplateforme ; elle est donc supportée par les environnements usuels. Il s'agit entre autres des éditions Windows, des systèmes Mac OS X et des différentes distributions Linux. Un simple travail de recompilation du code source sur l'environnement désiré permet de disposer d'une version compatible de l'application.

A.13 Déploiement

A.13.1 Installation de l'application

Le déploiement de l'application suit la procédure standard des environnements supportés :

- Pour un environnement Windows, le déploiement se réalise par le biais d'un installateur.
- Pour un environnement Mac OS X, le déploiement se réalise par le biais d'un fichier DMG.
- Pour une distribution Linux (Arch), le déploiement se réalise par le biais d'un paquetage.

A.13.2 Mise en place du matériel

La mise en place du matériel requis est décrite par la procédure suivante :

1. Disposez le support de travail.
2. Posez le stylet sur le support de travail.
3. Placez la caméra au-dessus du support de travail.
4. Placez le projecteur vidéo au-dessus du support de travail.

Les schémas suivants illustrent un environnement de travail idéal :

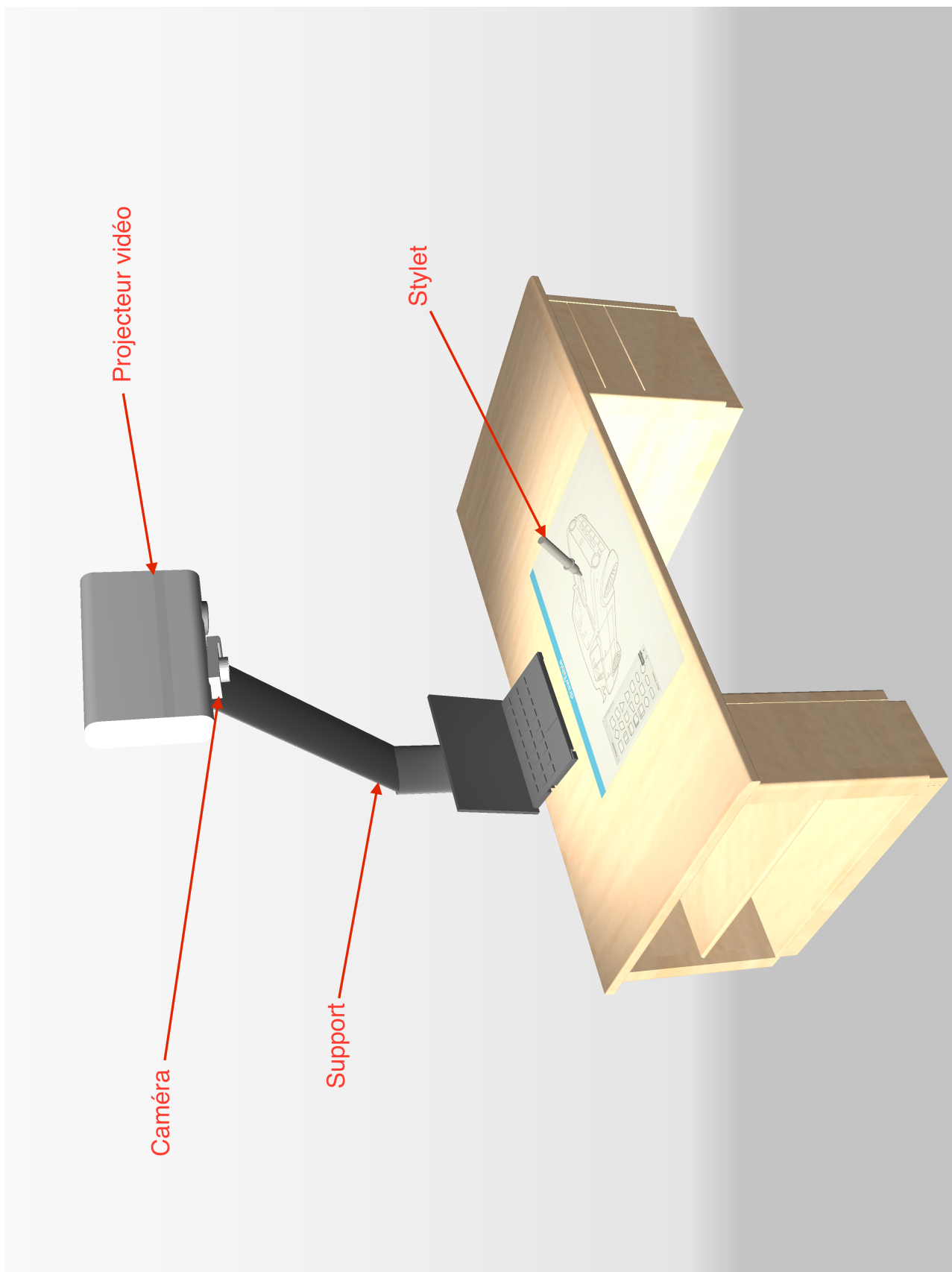


FIGURE A.3 – Vue en perspective d'un environnement idéal

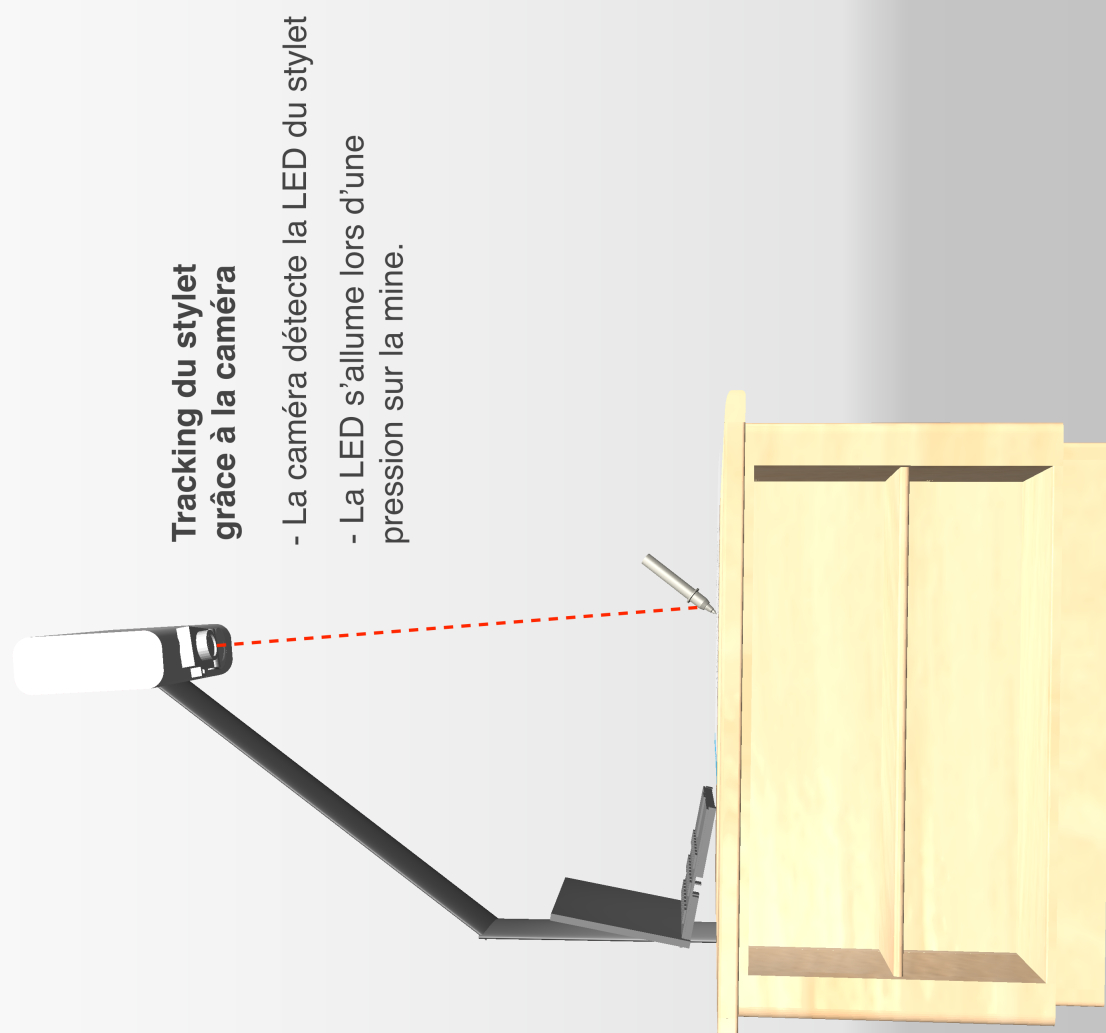


FIGURE A.4 – Vue de côté d'un environnement idéal

A.14 Déroutement du projet

A.14.1 Planification des tâches

Le diagramme de Gantt qui suit présente la planification des tâches du projet :

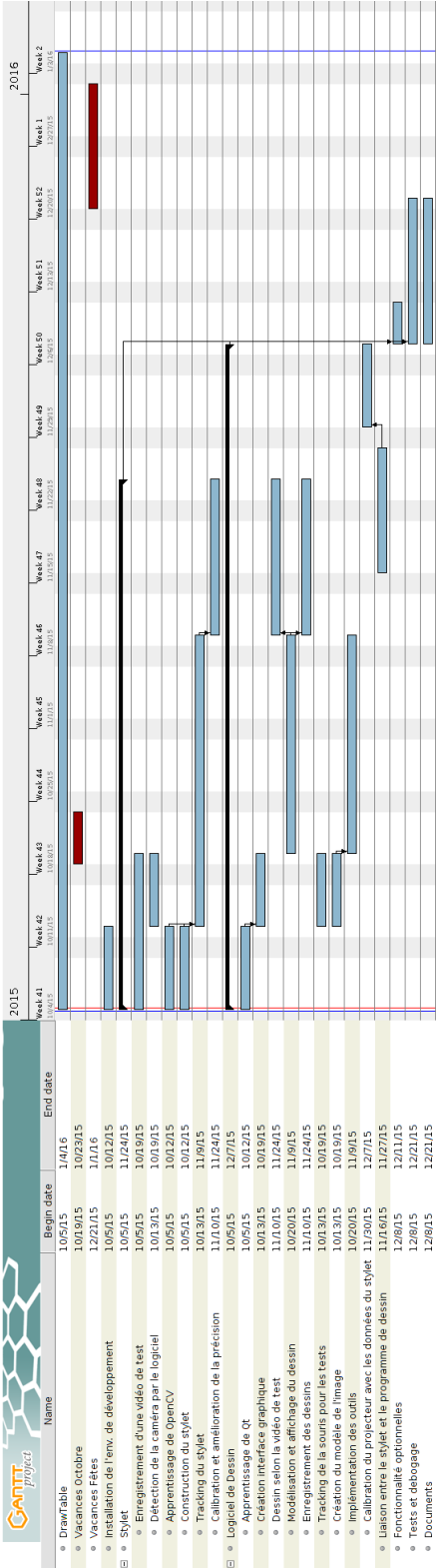


FIGURE A.5 – Planification initiale des tâches

A.14.2 Description des tâches

Les points suivants constituent un bref descriptif des tâches du projet :

- Enregistrement d'une vidéo de test
 - Afin de pouvoir tester les différents outils du logiciel de dessin avant que l'outil de tracking ne soit terminé, une vidéo contenant différents mouvements sera enregistrée.
- Apprentissage d'OpenCV
 - Pour toutes les opérations de tracking du stylet, nous allons utiliser la librairie OpenCV. Il sera donc nécessaire de prendre le temps d'en apprendre les bases avant de se lancer à corps perdu dans l'implémentation.
- Construction d'un stylet
 - Un stylet de notre propre création sera utilisé pour cette application, le matériel nécessaire à la fabrication de ce dernier est détaillé dans la description du prototype du stylet.
- Tracking du stylet
 - A l'aide d'une caméra, nous détecterons les mouvements du stylet, il faudra donc récupérer ces informations et les transmettre au logiciel de dessin.
- Apprentissage de Qt
 - Apprentissage des librairies Qt pour ce qui concerne principalement les interfaces graphiques et les librairies de dessin.
- Création de l'interface graphique
 - Conception de l'interface graphique du logiciel. Celle-ci contiendra un menu permettant d'importer/exporter des images, d'un plan de travail sur lequel travailler ainsi qu'une boîte à outils contenant différentes fonctionnalités telles que le crayon, la gomme, une palette de couleurs, le choix de l'épaisseur du trait.
- Tracking intermédiaire de la souris
 - Afin de permettre de tester le bon fonctionnement de l'implémentation des différents outils, ceci d'une manière accessible à tous les développeurs, il faut implémenter une fonctionnalité permettant de dessiner à l'aide de la souris.
- Création du modèle de l'image
 - Conception et création du modèle qui sera utilisé par le logiciel pour représenter notre image.
- Implémentation des outils
 - Implémentation des fonctionnalités obligatoires disponibles dans notre boîte à outils. Se référer à la description des fonctionnalités pour en connaître l'inventaire.
- Enregistrement des dessins
 - Exportation de notre représentation de l'image en une image au format .PNG.
- Modélisation et affichage du dessin
 - Implémentation des méthodes gérant l'affichage et la modélisation de notre image.
- Calibrage du projecteur avec les données du stylet
 - Calibrage permettant de faire concorder la position de l'image projetée avec la position effective du stylet.
- Dessin selon la vidéo de test
 - Utilisation d'une vidéo de test pré-enregistrée utilisant les méthodes de tracking implémentées

permettant de tester les différents outils sans pour autant avoir le matériel à disposition.

- Liaison entre le stylet et le logiciel
 - Liaison entre les données collectées par le système de tracking avec les méthodes d’affichage du logiciel.

Journal de travail

- B.1 Semaine 1 : 14 septembre 2015**
- B.2 Semaine 2 : 21 septembre 2015**
- B.3 Semaine 3 : 28 septembre 2015**
- B.4 Semaine 4 : 5 octobre 2015**
- B.5 Semaine 5 : 12 octobre 2015**
- B.6 Semaine 6 : 26 octobre 2015**
- B.7 Semaine 7 : 2 novembre 2015**
- B.8 Semaine 8 : 9 novembre 2015**
- B.9 Semaine 9 : 16 novembre 2015**
- B.10 Semaine 10 : 23 novembre 2015**
- B.11 Semaine 11 : 30 novembre 2015**
- B.12 Semaine 12 : 7 décembre 2015**
- B.13 Semaine 13 : 14 décembre 2015**
- B.14 Semaine 14 : 4 janvier 2016**
- B.15 Semaine 15 : 11 janvier 2016**

Annexe C

Planification

C.1 Planification initiale

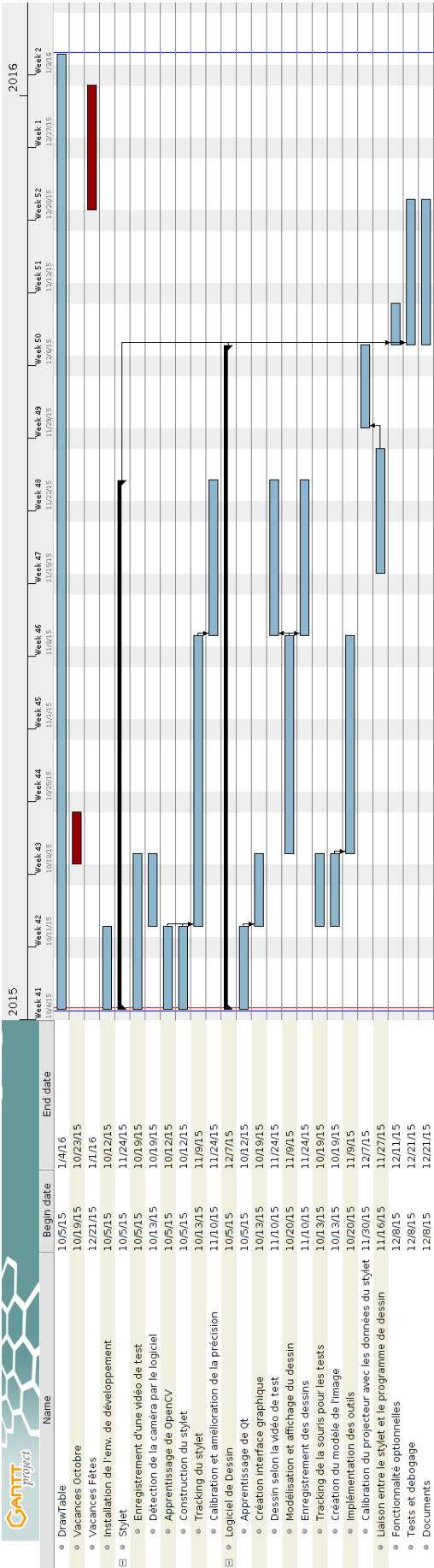


FIGURE C.1 – Planification initiale du projet

C.2 Planification finale

TODO : définir la planification finale du projet avec Gantt Project.