

|  |  |
| --- | --- |
| **CFI Montigny le Bretonneux** | |
| **Reconnaissance de pièces** | | |
| **Méthode Agiles** |

|  |
| --- |
| *Jean-Hilaire OBAME, Clément AUBRY, Marie LANOUE, Karam YELLES*  **21/12/2018** |



Sommaire

[1. Cadre du projet 3](#_Toc533166448)

[1.1. Méthode Agile : 3](#_Toc533166449)

[1.2. SCRUM 4](#_Toc533166450)

[1.2.1. Les sprints 4](#_Toc533166451)

[1.2.2. Le product backlog 4](#_Toc533166452)

[1.2.3. Mêlée quotidienne 5](#_Toc533166453)

[1.3. Contexte 5](#_Toc533166454)

[1.4. Objectif 6](#_Toc533166455)

[1.5. Organisation du projet 6](#_Toc533166456)

[2. Etude 8](#_Toc533166457)

[2.1. Trello 8](#_Toc533166458)

[2.1.1. Premier sprint : 10](#_Toc533166459)

[2.1.2. Deuxième sprint 10](#_Toc533166460)

[2.2. GitHub : 10](#_Toc533166461)

[3. Conception 12](#_Toc533166462)

[3.1. ArgoUML 12](#_Toc533166463)

[3.2. Acquisition d’image 14](#_Toc533166464)

[3.1. SolidWorks 14](#_Toc533166465)

[4. Réalisation 15](#_Toc533166466)

[4.1. Le logiciel Qt creator (en utilisant la bibliothèque OpenCV) 15](#_Toc533166467)

[4.2. Interface graphique 15](#_Toc533166468)

[4.2.1. Création de l’interface 1 15](#_Toc533166469)

[4.2.2. Création de l’interface 2 : 17](#_Toc533166470)

[4.2.3. Problèmes rencontrés 18](#_Toc533166471)

[4.3. Acquisition d’image 18](#_Toc533166472)

[4.3.1. Lecture de la caméra et capture de l’image depuis Qt 18](#_Toc533166473)

[4.3.2. Enregistrement de l’image dans un dossier 20](#_Toc533166474)

[4.3.3. Incidents rencontrés : 20](#_Toc533166475)

[4.4. Reconnaissance des pièces 21](#_Toc533166476)

[4.4.1. Limites 22](#_Toc533166477)

[4.4.2. Difficultés rencontrées 22](#_Toc533166478)

[5. Perspectives d’améliorations 23](#_Toc533166479)

[5.1. Support en impression 3D 23](#_Toc533166480)

[5.2. Interface administrateur 23](#_Toc533166481)

[5.3. Optimisation du programme d’identification des pièces de monnaies 24](#_Toc533166482)

[5.3.1. Détection des cercles 24](#_Toc533166483)

[5.3.2. Détection des types de pièce 24](#_Toc533166484)

[6. Conclusion 25](#_Toc533166485)

[7. Annexes 26](#_Toc533166486)

[7.1. Annexe 1 26](#_Toc533166487)

[7.2. Annexe 3 28](#_Toc533166488)

[7.3. Annexe 2 29](#_Toc533166489)

# Cadre du projet

Durant notre première année de master Manager des systèmes informatiques et robotiques, nous avons participé à différents projets dans un cadre pédagogique afin de nous préparer à nos futures responsabilités professionnelles.

Le projet qui fera l’objet de ce rapport est l’application d’une méthode de gestion de projet en mode « agile » sur une problématique bien définie.

## Méthode Agile :

Une méthode Agile est une approche interactive et collaborative, capable de prendre en compte les besoins initiaux du client et ceux liés aux évolutions du projet.

La méthode agile se base sur un cycle de développement qui porte le client au centre. Il est impliqué dans la réalisation du début à la fin du projet. Ainsi il peut obtenir une meilleure visibilité de la gestion des travaux.

L’implication du client dans le processus permet à l’équipe d’obtenir des retours réguliers afin d’appliquer directement les changements nécessaires. Cette méthode vise à accélérer le développement d’un logiciel. De plus, elle assure la réalisation d’un logiciel fonctionnel tout au long de la durée de sa création.

Le principe de base consiste à proposer une version minimale du logiciel puis à intégrer des fonctionnalités supplémentaires à cette base, par un processus de répétition. Ce processus regroupe une séquence d’instructions à répéter autant de fois que possible, selon le besoin.

La méthode agile repose sur quatre grands principes :

* **COLLABORATION** : La communication et la cohésion d’une équipe passent avant les outils et les processus.
* **EQUIPE** : Le privilège de la relation équipe/client est mis en avant plutôt que la négociation contractuelle.
* **APPLICATION** : Préférer une application bien construite à une documentation détaillée.
* **ACCEPTATION** : Le choix de l’acceptation du changement et de la flexibilité au détriment d’un plan rigide.

## ****SCRUM****

En effet, lorsque le terme « méthode agile » est employé au singulier on parle d’un concept, une approche. Cependant il existe plusieurs méthodes agiles qui se différencient les unes des autres.

De nos jours, la méthode « SCRUM » est la méthode agile la plus populaire. Cette méthode s’appuie sur des « sprints » qui sont des espaces temps pouvant aller de quelques heures pour les plus courts à un mois pour le plus longues. Généralement et de préférence un sprint s’étend sur deux semaines. A la fin de chaque sprint, l’équipe présente ce qu’elle a ajouté au client.

La méthode SCRUM regroupe trois acteurs :

* **Le Product Owner** ou **« Directeur de produit »** : Il communique les objectifs premiers des clients et des utilisateurs finaux, coordonne l’implication des utilisateurs et des parties prenantes et se coordonne éventuellement avec les autres product owners pour assurer une cohérence.
* **Le SCRUM Master** : Membre de l’équipe, il a pour but d’optimiser la capacité de production de l’équipe. Pour se faire, le SCRUM master aide l’équipe à travailler de façon autonome.
* **L’équipe** **opérationnelle** : Elle est composée idéalement de moins de dix personnes : la particularité d’une équipe SCRUM est qu’elle est dépourvue de toute hiérarchie interne. Une équipe SCRUM est autoorganisée.

L’avancement d’un projet SCRUM est rythmé par de grands principes permettant de garantir sa réussite. Ils sont détaillés ci-dessous.

### Les sprints

Le cycle de vie SCRUM est rythmé par des échéances de quelques semaines : les sprints.

A chaque début de sprint, une réunion de planification est organisée. Au cours de cette réunion, une sélection est faite dans les tâches du product backlog selon les exigences prioritaires pour le client. Elles seront développées, testées et livrées au client à la fin du sprint.Elles constituent le **sprint backlog, un sous ensemble du product backlog.**

### Le product backlog

**C’est un référentiel des exigences initiales**, dressé et hiérarchisé avec le client. Il constitue ce que l’on nomme le **product backlog.** Il ne doit pas nécessairement contenir toutes les fonctionnalités attendues dès le début du projet, il peut évoluer durant le projet en fonction des besoins du client.

### Mêlée quotidienne

Il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait au début de chaque journée. Elles ont une durée de 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions : « Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ? »

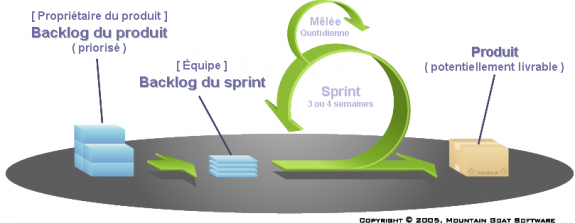


Figure 1 : Schéma du processus SCRUM

## Contexte

C’est en partant de cette feuille de route que nous avons appliqué la méthode de gestion de projet SCRUM afin de réaliser le dessein qui nous a été attribué.

Prévoyant de récolter un nombre conséquent de pièces de monnaies à l’approche des fêtes de fins d’années et dans le but de simplifier le comptage de monnaies par leurs bénévoles, une association a décidé de faire appel aux étudiants de Master 1 informatique et robotique du CFI Montigny le Bretonneux afin de trouver une solution à leur problème. Le but étant de concevoir et de mettre en place un logiciel facilitant la reconnaissance et le comptage de pièces de monnaies. Le produit devra être opérationnel et prêt à l’utilisation à partir du weekend précédant les fêtes de noël (à partir Du 22 décembre 2018).

## Objectif

Ce projet comporte plusieurs objectifs :

* Livrer un produit prêt à l’utilisation et ce dans les temps impartis par le client (produit prêt à être utiliser pour le 22 Décembre).
* Concevoir un produit simple d’utilisation et ludique afin que la prise en main du produit soit la plus intuitive possible pour le client.

## Organisation du projet

Afin de répondre au mieux à la demande de notre client ainsi qu’au fondamentaux de la méthode SCRUM, nous avons mis en place une équipe composé de 4 membres :

* LANOUE Marie : Qui a le rôle de développeur ainsi que celui du **SCRUM Master**
* AUBRY Clément : Qui a le rôle de développeur ainsi que celui de **Product Owner**
* OBAME NDONG Jean-Hilaire : Qui a le rôle de Développeur
* YELLES Karam : Qui a le rôle de Développeur

Afin de répondre au mieux aux besoins de notre client, nous avons décidé d’effectuer une analyse « QQOQCCP » sur notre problématique.

C’est une méthode de résolution de problème permettant d’avoir toutes les dimensions d’une problématique afin d’obtenir les informations élémentaires suffisantes pour identifier ses aspects essentiels de cette dernière. Cette méthode se réalise en répondant de manière précise à six questions :

* Quoi ? *« De quoi s’agit-il ? »*
* Qui ? *« Qui est concerné ? »*
* Où ? *« Où cela se produit-il et s'applique-t-il ? »*
* Quand ? *« Depuis quand vous avez ce problème ? »*
* Combien ? *« Combien de fois ce problème se produit-il ? »*
* Comment ? *« Comment se produit le problème ? »*
* Pourquoi ? *« Pourquoi le problème se pose ? »*

Figure 2 : QQOQCCP du projet de reconnaissance de monnaie

Suite à cette analyse, nous avons pu établir une feuille de route qui nous servira de fil conducteur à la bonne réalisation de notre mission. Cette feuille de route nous servira de base à la construction et à la gestion de notre projet tout en appliquant les méthodologies SCRUM apprises en formation.

# Etude

Afin de réaliser au mieux notre projet, nous avons utilisés différents outils tel que :

## Trello

C’est un outil de gestion de projet en ligne basé sur une organisation des projets en colonnes listant des cartes représentant des tâches. Chaque tâche peut être assignée à différents membre d’une équipe. Les tâches peuvent être déplacées d’une colonne à l’autre, traduisant leur avancement. La conception de ce site est basée sur le fonctionnent des méthodes agiles et tout particulièrement à la méthode SCRUM.

C’est un site gratuit permettant à chaque membre de l’équipe d’y accéder et de pouvoir le modifier afin de le mettre à jour à tout moment.

Nous avons donc utilisé Trello afin de mettre en place un planning SCRUM à l’intérieur duquel nous avons développé et réparti les tâches utiles à notre projet. Les différentes tâches ont été hiérarchisées, répartie en colonnes puis attribuées à différents membres de l’équipe projet. Nous avons trouvé judicieux de séparer notre projet en deux « sprints » afin d’avoir un meilleur contrôle sur l’avancement de notre projet. Ceci nous permet également de pouvoir présenter les avancées du projet au client, ce qui nous permettra d’obtenir des feedbacks constructifs de mi-projet afin que nous puissions apporter les corrections nécessaires à notre logiciel.

Pour finir, nous avons estimé un délai de réalisation pour chacune des tâches du projet en cohérence avec le délai de deux semaines en cohérence avec notre client.

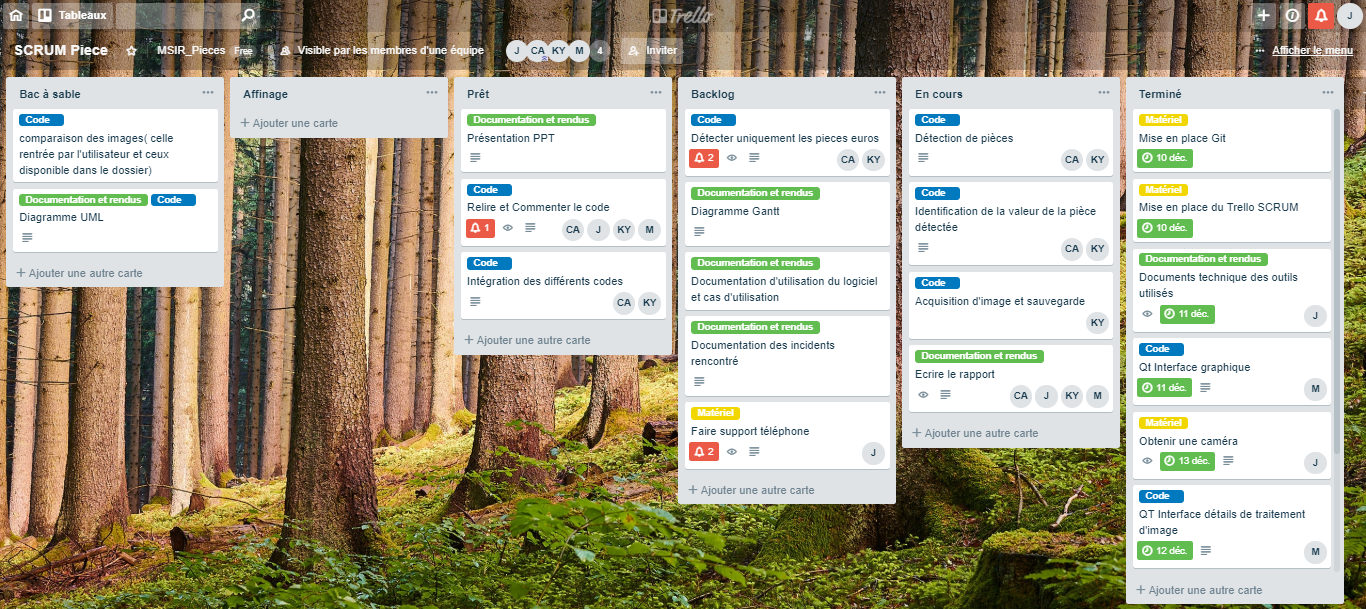


Figure 3 : Tableau de planification des tâches du projet reconnaissances des pièces

Les tâches ont été divisées en deux sprints d’une semaine :

### Premier sprint :

* Mise en place des outils du projet (Trello, GitHub, QT creator ainsi que la bibliothèque OpenCV)
* Rédaction du rapport (Document technique, Digramme de Gantt, Diagramme UML).
* Conception de l’interface graphique du logiciel sur Qt
* Programmation (Affichage du contour des pièces)
* Obtention d’une caméra permettant la détection des pièces
* Programmation (acquisition et sauvegarde d’image)
* Programmation (détection et identification de la valeur des pièces détectées)

### Deuxième sprint

* Réalisation du Power Point
* Finalisation du rapport (document d’utilisation du logiciel, document d’incident rencontré, diagramme d’utilisation)
* Conception du support caméra
* Programmation (Détection unique des pièces d’euros puis faire leurs sommes)
* Mise en relation de différentes parties du code ainsi que l’interface graphique
* Relecture et commentaire du code

Le travail réalisé sur ce logiciel ainsi que le délai de livraison du produit imposé par notre client nous a permis de mettre en place un diagramme de GANTT (page 11)

## GitHub :

Pour ce projet nous avons mis en place un système d’archivage de données afin que chaque membre du groupe puisse se transmettre différents documents utiles dans notre progression (fichiers, dossiers utiles au projet). Nous avons utilisé un logiciel de gestion des versions nommé GitHub.

C’est un logiciel permettant de stocker différents documents sur une plateforme. Il utilise le logiciel de gestion de version GIT. Il permet d’archiver les différentes versions d’un programme.

Ce logiciel nous permet d’accéder aux travaux liés aux tâches gérées par chaque membre de l’équipe. Il nous offre aninsi la possibilité d’apporter des ajustements éventuels ou des commentaires pertinents.



Figure 4 : Diagramme de GANTT du projet de reconnaissance des pièces

# Conception

Avant de nous lancer dans la programmation pure de notre futur logiciel, nous avons décidé de mettre en forme de façon schématique son diagramme de classe afin de pouvoir visualiser les interactions possibles entre les différentes interfaces de notre logiciel. Puis un diagramme de cas d’utilisation dans le but d’obtenir une vision globale du comportement fonctionnel du logiciel.

## ArgoUML

C’est un logiciel libre spécialisé dans la création de différents types de diagramme UML (diagramme de cas d’utilisation, de classes, de séquence, d’état, de collaboration, d’activité et de déploiement).

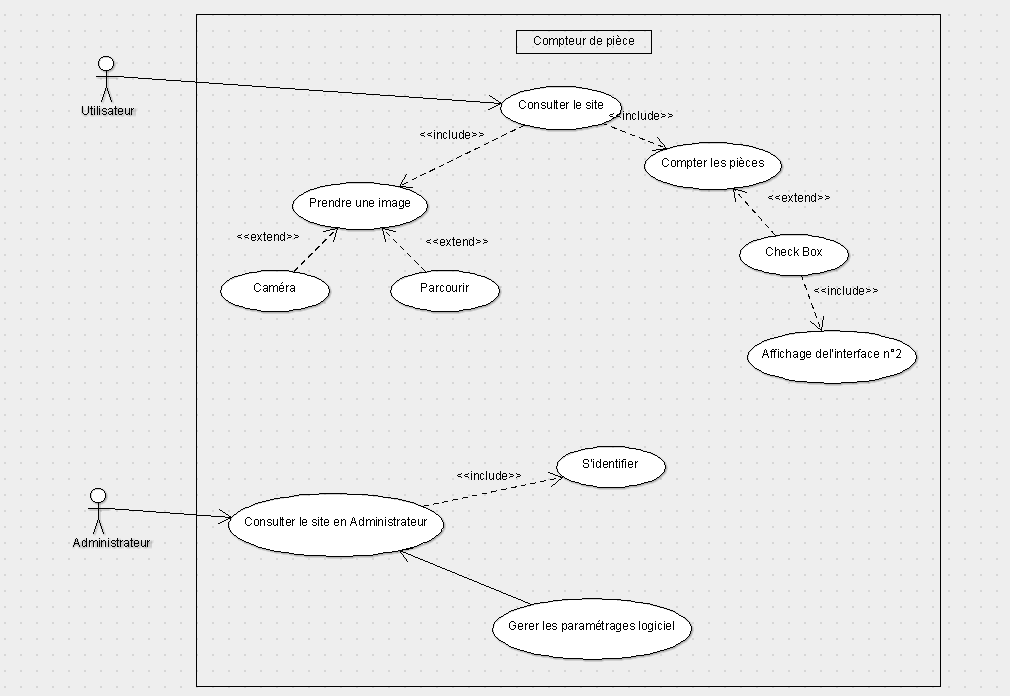
****

Figure 5 : Diagramme des cas d’utilisation du logiciel

Le diagramme des cas d’utilisation ci-dessus permet de visualiser de façon générale les différents cas d’utilisation des différentes personnes qui auront accès au logiciel.

Nous avons donc identifié deux types d’utilisateurs pour le futur logiciel :

* **Utilisateur lambda** : Aura accès aux fonctions d’utilisation classique du logiciel :
  + Application générale de comptage de pièces
  + Prise d’une image
  + Comptage des pièces par image
  + Affichage de l’interface détaillée du traitement d’image
* **Administration :** Aura accès à toutes les fonctions d’utilisations classique du logiciel et pourra accéder à la modification des paramètres du logiciel :
  + Application générale de comptage de pièces
  + Prise d’une image
  + Comptage des pièces par image
  + Affichage de l’interface détaillé du traitement d’image
  + Gérer les paramètres du logiciel

Chaque fenêtre d’affichage dans Qt est gérée par une classe. L’ensemble des traitement d’image sont gérés dans la fenêtre principale.

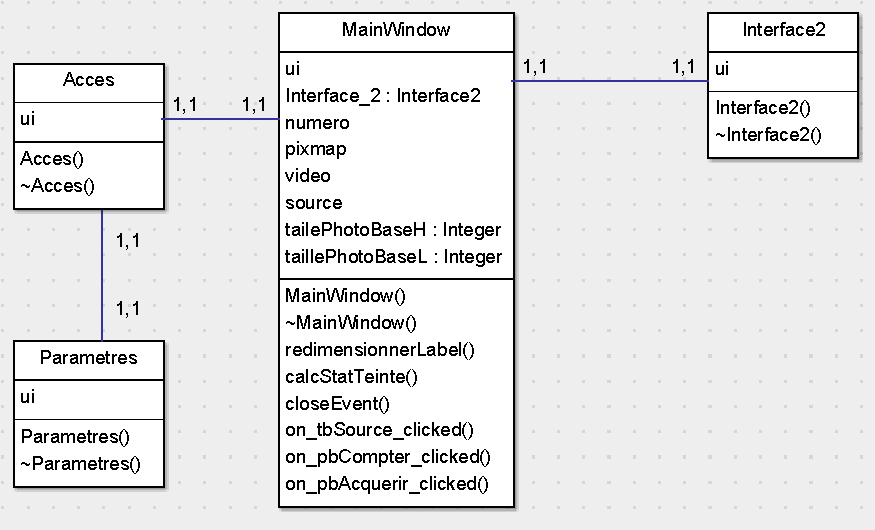


Figure 6 : Diagramme de classes du logiciel

## Acquisition d’image

Il existe différents moyens d’acquérir une image de pièce en temps réel afin de pouvoir la traiter par la suite (appareil photo, caméra, téléphone portable, webcam, …). La solution qui nous paraissait la plus pratique et la plus économique pour notre client était l’utilisation d’une webcam pour effectuer l’acquisition d’une photo.

Avantage : Simple d’utilisation, facile d’acquisition, simple de paramétrage sur Qt creator.

Afin d’utiliser de façon optimale ce type de webcam, il fallait concevoir un support permettant à la fois de soutenir notre outil d’acquisition d’images et de pouvoir marquer la délimitation d’une zone de capture d’image des pièces à traiter. Pour cela nous avons décidé de concevoir, sur un logiciel CAO, un support alliant les deux fonctions.

## SolidWorks

Solidworks et un logiciel CAO (Conception Assisté par Ordinateur) de type propriétaire fonctionnant sur Windows. Ce logiciel fonctionne sur la mise en relation de 3 types de fichiers relatifs à trois concepts de base : la conception d’une pièce, d’un assemblage et leurs mises en plan. Ces fichiers sont en relation. Toutes modifications à quelque niveau que ce soit sont répercutées vers tous les fichiers concernés. Pour notre besoin, nous avons utilisé deux des trois concepts de base du logiciel :

* Conception d’une pièce : Support Webcam (dessin ci-dessous)
* Mise en plan de la pièce : Support Webcam (Présenté en annexe)

La Webcam sera positionnée et fixé grâce à un système de serrage étau. Ce système d’étau se situera à une vingtaine de centimètre de la zone de positionnements des pièces à traiter. La zone d’acquisition des pièces sera délimitée par le cercle situé à la base de notre support. Afin que la webcam capture de façon optimale les images des pièces à additionner, le point d’origine des deux cercles (support Webcam et le socle du support) a été aligné sur le même axe verticale.

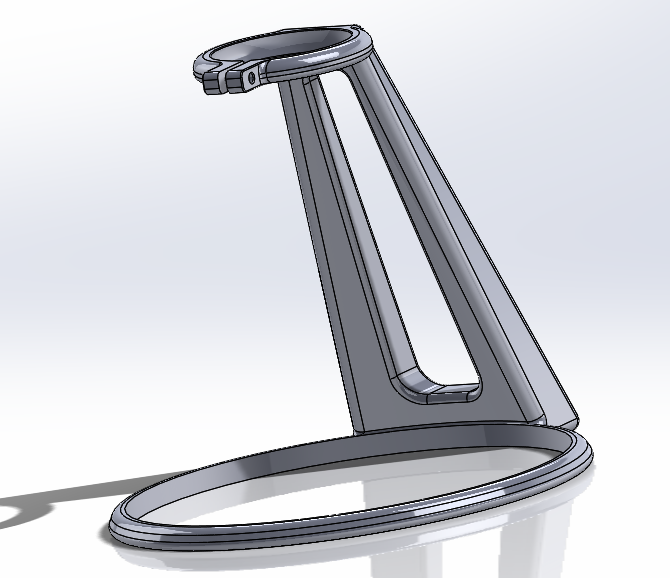


Figure 7 : Dessin de conception du support caméra

# Réalisation

## Le logiciel Qt creator (en utilisant la bibliothèque OpenCV)

Afin de réaliser notre application, nous avons utilisé le logiciel Qt creator. En effet ce logiciel offre une interface graphique permettant de configurer un visuel sur notre application de reconnaissance de pièce. C’est un logiciel orienté objet et développé en C++. Notre préférence s’est portée sur ce logiciel car il répondait au mieux à notre besoin. C’est un logiciel avec lequel nous avons pu nous familiariser au cours de notre formation et qu’il nous paraissait pertinent d’approfondir.

Afin de nous permettre d’exploiter au mieux les capacités du logiciel Qt, nous avons décidé d’installer la bibliothèque Open CV (Open Computeur Vision). C’est une [bibliothèque graphique](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_graphique) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre), spécialisée dans le [traitement d'images](https://fr.wikipedia.org/wiki/Traitement_d%27images) en temps réel.

## Interface graphique

### Création de l’interface 1

L’interface devra pouvoir récupérer l’image à traiter et attribuer cette image à un label voulu (pour l’affichage des pièces)

Fonction utiliser pour prendre une photo et l’attribuer à un label :

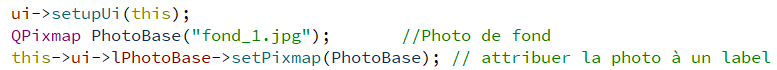
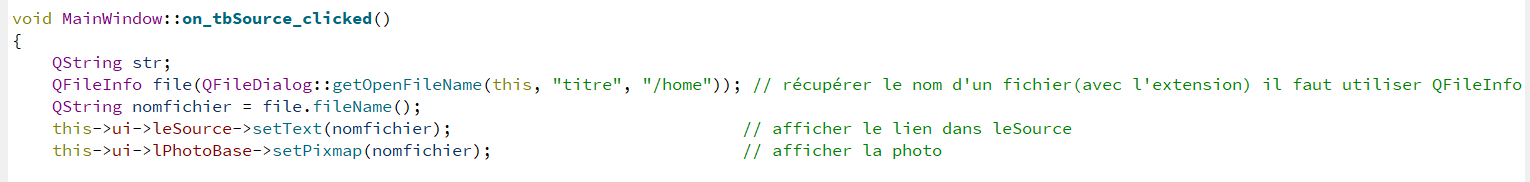




Figure 8 : Ecran principal du logiciel

Nous avons créé une touche parcourir pour aller chercher une image dans un dossier et afficher le nom du fichier sélectionné dans le label correspondant. Nous avons également jugé important que les futurs utilisateurs puissent visualiser le nombre de pièces de chaque type.

Voici la fonction utilisée pour la touche parcourir de l’interface. Elle permet d’aller chercher une photo dans un dossier et afficher le lien source de la photo dans le label correspondant :



### Création de l’interface 2 :

Nous avons mis en place un deuxième écran permettant de visualiser plus en détail les filtres utilisés dans le traitement de l’image.

Ci-dessous le code qui permet d’accéder à l’interface 2, ainsi que les touches correspondantes :

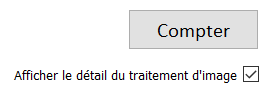
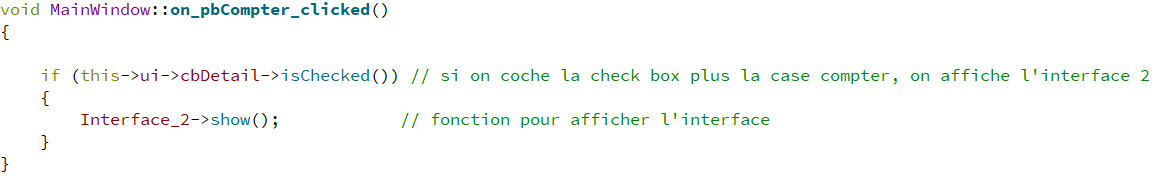


Figure 9 : Bouton Compter



Une fois la checkbox cochée, appuyer sur l’icône « compter » affiche l’interface suivant :



Figure 10 : Ecran d’affichage des traitements d’image

Cette fenêtre sert à détailler le traitement appliqué sur l’image choisie par l’utilisateur et sert aux développeurs à vérifier si le traitement de l’image a été réalisé correctement.

### Problèmes rencontrés

* Faire communiquer les deux interfaces 🡪 faire en sorte que l’interface 2 s’affiche seulement lorsqu’on coche la check box et que l’on appuie sur compter grâce à une boucle if.
* Dimensionner l’image à la taille du label.

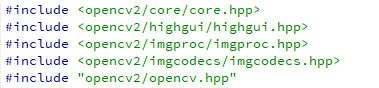
Après avoir réalisé le plan d’interface sur Qt, nous allons à présent procéder à la partie Acquisition d’image par la caméra depuis notre interface.

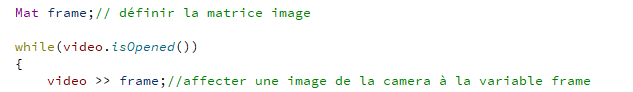
## Acquisition d’image

Cette partie est primordiale pour notre application, elle permet d’obtenir l’image des pièces de monnaie depuis la caméra, afin d’effectuer le traitement et identifier les pièces et ainsi en déduire la somme.  
Le matériel utilisé dans cette partie est une Webcam Logitech 150x

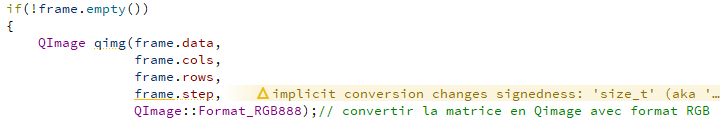
### Lecture de la caméra et capture de l’image depuis Qt

Pour pouvoir accéder à la caméra depuis l’interface Qt, il faut tout d’abord inclure des bibliothèques d’openCV, qui permettent d’utiliser des objets dédiés à l’imagerie. Ces bibliothèques sont les suivantes :

  
  
Sachant que la vidéo est un ensemble d’image (frames), et que l’image est une matrice, nous avons défini une variable frame de type « Mat » (un objet d’openCV) pour stocker l’image capturée depuis la caméra.

Ensuite nous avons converti cette matrice en QImage afin de pouvoir l’afficher sur la fenêtre de Qt. Et comme open CV traite les images en format BGR, nous avons dû la convertir en format RGB pour l’afficher avec les bonnes couleurs sur notre interface.



Et enfin nous avons configuré la taille de l’image pour qu’elle ait la même taille que celle notre fenêtre.



Voici une illustration du résultat obtenu :

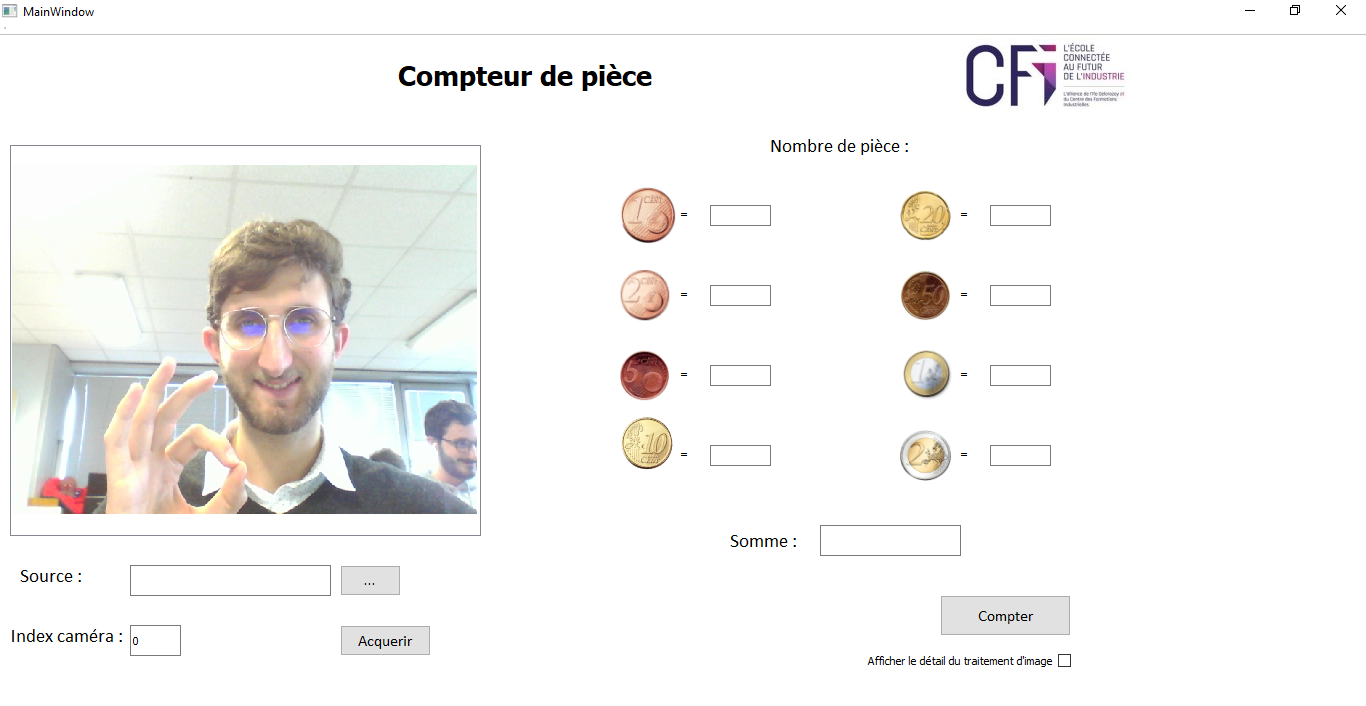
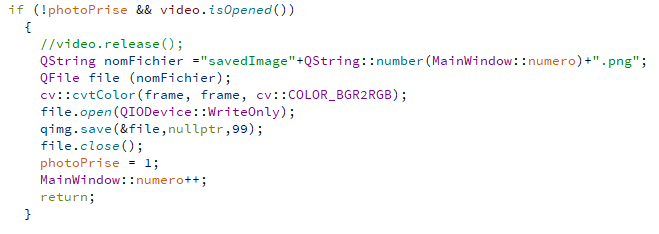


Figure 11 : Lecture de la caméra depuis l’interface Qt

### Enregistrement de l’image dans un dossier

Une fois qu le bouton acquérir est cliqué, l’image contenue dans la variable de type QImage, est enregistrée sous format « PNG » dans un dossier spécifique sur le disque dur. Un numéro est incrémenté et affiché à côté du nom du fichier pour que l’ancienne capture ne soit pas écrasée par la nouvelle.  
  


Pour voir plus en détail les étapes et les démarches suivies, veuillez trouver en annexe le code complet.

### Incidents rencontrés :

* Enregistrement d’un fichier vide dans le dossier spécifié
* Sauvegarde de plusieurs images lors d’un seul clic.

## Reconnaissance des pièces

Afin de détecter et identifier les pièces, nous avons étudié et testé différents outils et méthodes, notamment :

* Conversion de l’image en niveaux de gris.
* Application de filtres gaussien et médian.
* Détection des contours avec les filtres de Sobel et Canny.
* Détection des cercles par la transformée de Hough.
* Mise à l’échelle de l’image étudiée.
* Extraction d’une partie de l’image pour étude individuelle.
* Calcul de l’histogramme et de sa moyenne.
* Extraction de la teinte et de la saturation d’un fragment image, avec calcul de leur moyenne et variance.
* Egalisation de l’histogramme par application d'un CLAHE (Contrast Limited Adaptive Histogram Equalization), avec conversion de l’image de RGB (Rouge vert Bleu) à LAB (Lightness Green-Red Blue-Yellow).
* Mise en place des données d’entrainement pour un algorithme de machine Learning.

L’algorithme de reconnaissance des pièces final se divise selon les étapes suivantes :

1. Détection des cercles
2. Calcul du diamètre des cercles
3. Détermination des données de couleur
4. Attribution des labels des pièces en fonction de leur diamètre et de leur couleur

La détection des cercles se fait par la transformée de Hough circulaire, sur une image en niveau de gris, floutée pour améliorer la détection.

L'algorithme de la transformée de Hough est composé de trois étapes :

* Pour chaque point de contour détecté, détermination de la courbe correspondante dans l'espace des paramètres ;
* Construction de la matrice d'accumulation à partir de ces courbes ;
* Détection des pics dans la matrice d'accumulation pour déterminer le cercle le plus pertinent.

Les paramètres de base à modifier dans cette transformée pour adapter la détection des cercles à l’image sont les suivants : distance minimale entre les centres, rayons minimum et maximum des cercles détectés.

La transformée nous fournit les centre et rayon des cercles détectés, ce qui nous permet de la comparer entre eux.

Afin d’analyser la couleur, nous devons convertir l’image du format RGB en HSV (Hue Saturation Value ou Teinte Saturation Luminance) afin d’accéder aux valeurs de Teinte pour la couleur et de Saturation pour l’intensité de la couleur.

La teinte permet de distinguer les pièces de cuivre (rouges) des autres pièces (jaunes).

La saturation nous permet de différencier les pièces de 1euro, dont le centre est gris, donc moins saturé.

### Limites

Détection de la couleur

Dans cette version du programme, la comparaison de couleur s’effectue sur la moyenne des pixels au sein du cercle détecté. La limite étant que selon l’éclairage, les valeurs de teinte et de saturation varient.

Détection des contours

Dans cette version du programme, les paramètres de la transformée de Hough sont ajustés à la main, de manière empirique. Cependant, selon l’image, son cadrage, et son échelle, ces paramètres doivent varier pour optimiser la détection des cercles

Nous aborderons les moyens de dépasser ces limites dans la partie 5 « Perspectives d’améliorations ».

### Difficultés rencontrées

Connaissance des outils :

Apprendre la syntaxe opencv, perfectionner les syntaxes C++ et Qt. De plus, un nombre non négligeable de documentation pour opencv se fonde sur le Python et nécessite donc une traduction en C++ pour être exploitée.

Connaissance du traitement d’image

Déterminer les méthodes les plus adaptées à notre objectif et en comprendre les principes sous-jacents.

Gestion des types de variables

Parvenir à interfacer les types de C++, Qt et opencv a été à l’origine d’un nombre conséquent de délais, en perturbant l’affichage, l’accès aux données et les calculs impliquant différents types.

# Perspectives d’améliorations

D’un point de vue global notre application correspond à ce que nous souhaitions réaliser. Le nombre limité de sprints laisse cependant la place à plus d’optimisation, en mener lors de prochains sprints.

Nous avons identifié plusieurs axes d’améliorations de notre programme, détaillés ci-dessous.

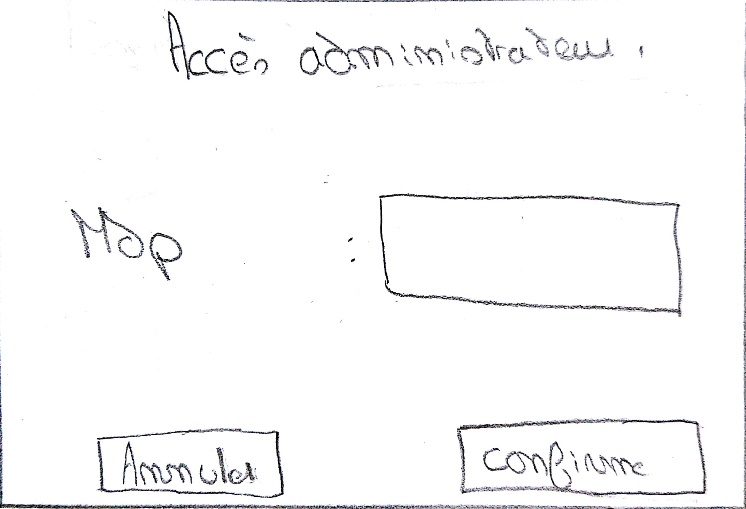
## Support en impression 3D

La mise à disposition d’un modèle 3D prêt à imprimer permet de normaliser la distance de prise de vue et améliore le confort d’utilisation de l’application, aux dépends de sa portabilité. Son utilisation serait optionnelle. Inclure un éclairage à ce support serais également un bon moyen d’améliorer la détection des pièces.

Voir Annexe 2 pour le plan

## Interface administrateur

L’ajout d’une page administrateur permet de donner accès à certains paramètres de l’application, pour des personnes habilitées.



**Paramètres à modifier :**

* Distance entre les centres des pièces
* Diamètres des pièces minimale / maximale
* Echelles des photos.

Figure 12 : Maquette de la fenêtre Acces

Les paramètres cités ici sont pertinent dans l’état actuel de l’application mais nécessiteront un ajustement dans les futures itérations

## Optimisation du programme d’identification des pièces de monnaies

### Détection des cercles

Afin d’améliorer la détection des contours des pièces, nous envisageons de segmenter l’image pour traiter les pièces une à une. Ainsi, nous pourront automatiser la détection des paramètres optimaux de la transformée de Hough pour chaque pièce et obtenir un diamètre plus précis. La comparaison des diamètres sera donc plus pertinente.

### Détection des types de pièce

Cet aspect de l’optimisation est double. D’une part permettre d’identifier les euros au milieu d’autres pièces, d’autre part de mieux reconnaître les différentes couleurs de pièce parmi les euros, notamment pour pouvoir gérer les différences d’éclairage.

Dans ce but, nous envisageons de mettre en place un algorithme de machine Learning. En se fondant sur une banque de données de pièces photographiées sous différents éclairages, l’algorithme aura pour fonction de déterminer la couleur d’une pièce (2€, 1€, pièces de bronze et pièces de cuivre). Cette méthode nous permettra de nous affranchir des variations d’éclairage.

# Conclusion

Après deux semaines de projet de reconnaissance de pièces en appliquant la méthode agile, nous avons fait en sorte de répondre au mieux aux besoins de notre client. Bien que le temps imparti fût assez cours, nous avons pu nous répartir les tâches de façon judicieuse selon les compétences de chacun afin de pouvoir tous avancer vers le même objectif.

Cette expérience fut très enrichissante pour nous à plusieurs points de vue :

* **Gestion de projet :**
  + Pouvoir travailler en équipe de façon agile nous à permit de gagner en autonomie dans notre prise de décision. Suivre de façon rigoureuse un planning et se tenir aux délais préconisés.
  + Apprendre à répartir des tâches selon les compétences de chacun des membres d’une équipe.
  + Réaliser un projet en appliquant une méthodologie de gestion de projet (SCRUM).
  + Appliquant des outils de gestion de projet tel que TRELLO, GitHub, diagramme de Gantt.
* **Technique :**
  + Amélioration de nos compétences en programmation (C++, Qt).
  + Familiarisation avec la bibliothèque OpenCV.
  + Utilisation de logiciel tel que le logiciel ArgoUML ainsi que SolidWorks.
  + Connaissance personnelle en traitement d’image ainsi que dans l’utilisation de différents types de filtres.

L’une de nos plus grandes satisfactions durant toute la durée de ce projet est la concentration et l’implication dont nous avons su faire preuves, malgré les perturbations externes. Ceci nous servira grandement lors de nos futurs projets, autant sur le plan personnel que professionnel.

# Annexes

## Annexe 1

Documentation d'utilisation du logiciel

Préconisation pour l’utilisation de notre logiciel :

* Utiliser un fond noir derrière les pièces ;
* Disposer les pièces de manière uniforme et les espacer en évitant de les coller ;
* Ne mettre que des Euros ;
* Privilégier les photos de bonne qualité.

Afin de faire fonctionner le logiciel, il faut lui fournir une photo. Pour ce faire il existe deux méthodes :

* Utiliser une image qui se trouve sur votre ordinateur dans le dossier *build* du programme, que vous allez chercher via la touche parcourir présent sur l’interface.
* Brancher une caméra puis utiliser les touches de l’interface pour capturer ce que voit la caméra.

Une fois la photo prise et affichée sur l’interface, cliquez sur la touche « Compter » pour que le programme compte les différentes pièces sur la photo et affiche sur l’interface le résultat dans la case « Somme ».



1

6

7

8

9

2

3

5

4

Figure 13 : Vue d’ensemble de l’interface

**Légende :**

* 1. Affichage des différentes photos sélectionnées par l’utilisateur.
  2. Touche parcourir pour aller chercher une photo dans un dossier le dossier BUILD.
  3. La source de la photo s’affichera ici.
  4. Index de la caméra :

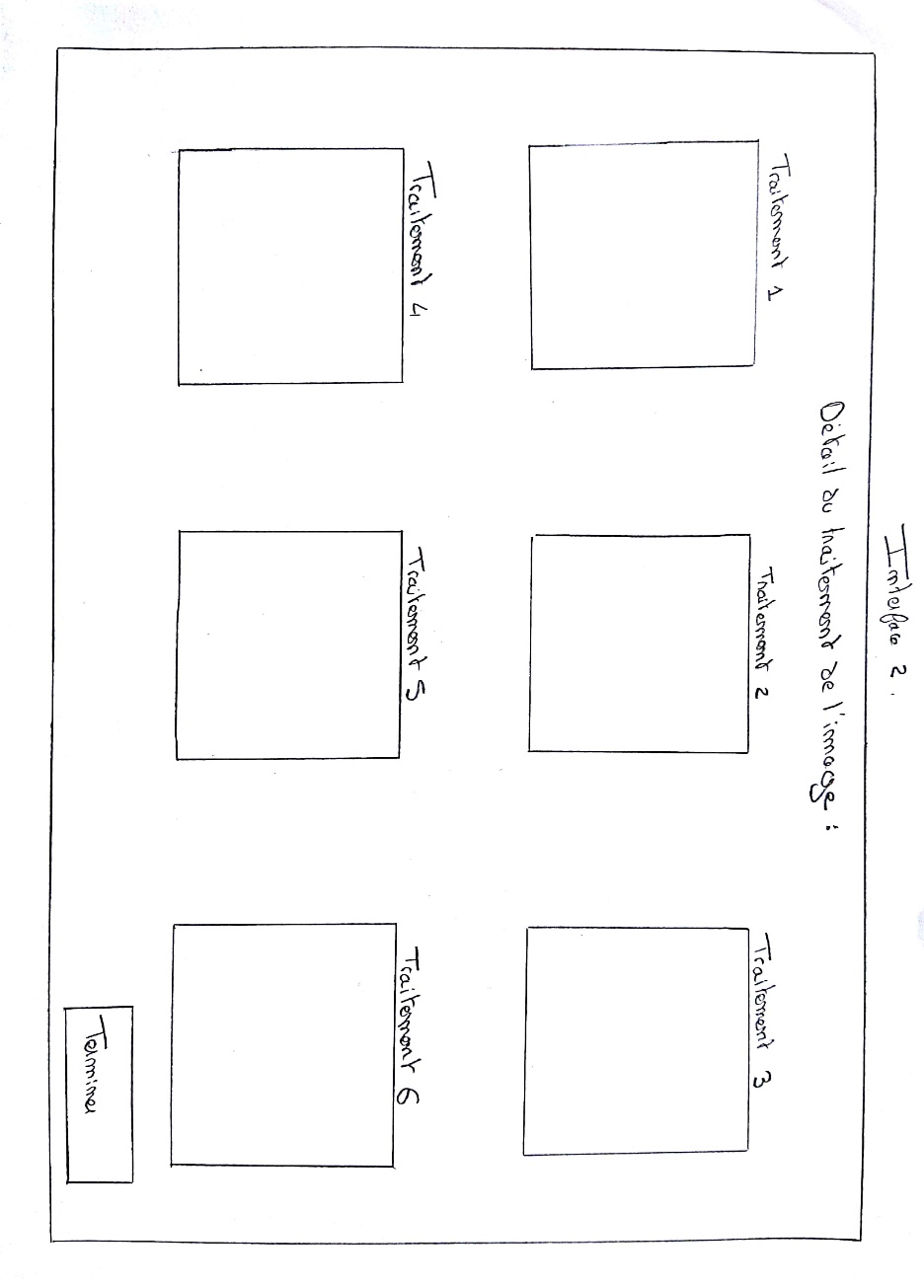
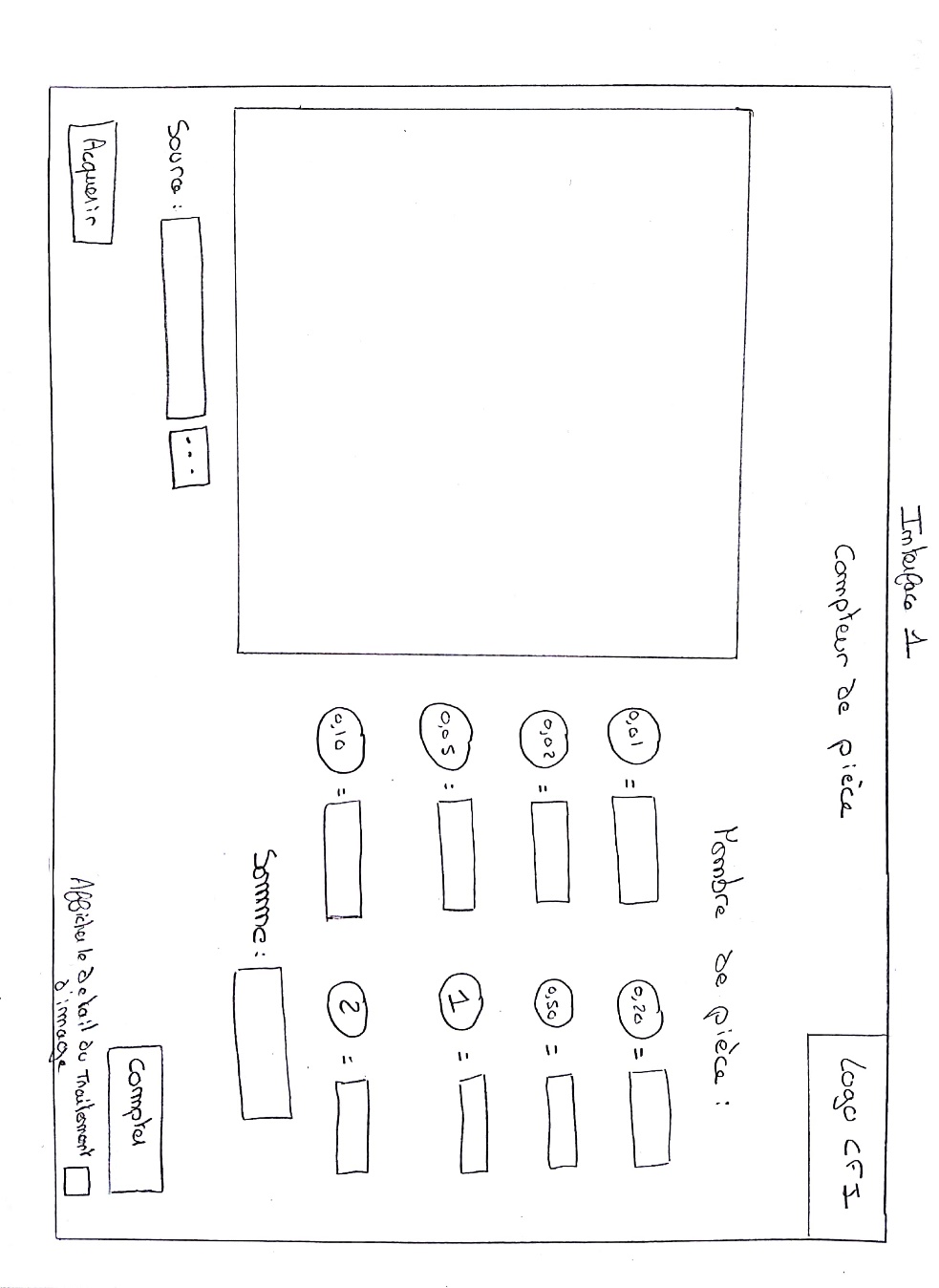
0 🡪 webcam de l’ordinateur portable

1 ou plus 🡪 caméra supplémentaire

* 1. Touche pour acquérir une image de la caméra.
  2. Interface où sera affiché le nombre de pièce par type de pièce.
  3. La somme de toutes les pièces présentes sur l’image choisie par l’utilisateur.
  4. Touche Compter pour lancer le programme et compter les pièces.
  5. Check Box à cocher si l’utilisateur souhaite afficher le détail du traitement fait sur l’image.

## Annexe 3

Figure 14 : Maquette des interfaces



## Annexe 2

Figure 15 : Plan de conception du support Webcam

