



Toujours s'assurer que les logos utilisés sont bien les bons et la version à jour ...
Ici, ce n'est pas le cas ...

Master 2 SIA
Parcours XX

adresse où la formation
peut être contactée (ex. :
adresse du secrétariat)

Adresse
société :

Xx
7, route de la
plage
.....
LOGO société

Ajuster les champs du tableau selon besoins

RAPPORT de PROJET/STAGE

Mini-Projet de techniques de Programmation avec mise en œuvre en Langage C++

C++2018 : Gestion de bibliothèques de fichiers signaux/images et traitements de ces fichiers

XXX 2018 – XXX 2019

Nom(s) du (des) auteur(s) du rapport

XXX

La page de garde est comme une "carte de visite" : celui qui le prend en main doit pouvoir immédiatement identifier :

- ⇒ la nature du document : rapport de stage, document technique, ...
- ⇒ l'auteur et son appartenance (formation et/ou société) + le lieu de réalisation du travail + les adresses pour pouvoir s'y adresser
- ⇒ la durée et l'année
- ⇒ le sujet bien évidemment
- ⇒ le ou les responsables du travail, pour pouvoir les contacter ou simplement savoir s'il les connaît

----- SAUT DE PAGE : À METTRE CHAQUE FOIS QUE NÉCESSAIRE POUR QUE LA PAGE DE
TEXTE SUIVANTE SE TROUVE PLACÉE SUR UNE PAGE DE DROITE, DANS L'OPTIQUE D'UNE
IMPRESSION RECTO/VERSO -----

Remerciements¹

Je tiens tout d'abord à remercier Monsieur / Madame **XXX**, Directeur de la société **XXX**, pour son accueil au sein de **XXX** à Toulouse.

Je remercie ensuite mon maître de stage, Monsieur / Madame **XXX**, de m'avoir permis d'effectuer ce stage au sein de l'équipe **XXX** et de m'avoir encadré tout au long du projet.

....

Je remercie enfin toutes les personnes que j'ai pu rencontrer pendant ce stage, pour leur accueil chaleureux et leurs divers conseils.

----- Saut de page -----

Sommaire²

LISTE DES FIGURES	1
LISTE DES ANNEXES.....	1
GLOSSAIRE	1
INTRODUCTION	2
CHAPITRE 1 : PRÉSENTATION DE L'ENTREPRISE	3
CHAPITRE 3 : PRÉSENTATION DU SUJET	
I. PROBLÉMATIQUE.....	
II. TECHNOLOGIES MISES EN ŒUVRE	
III. ORGANISATION	
CHAPITRE 4 : MISE EN ŒUVRE DU PROJET	
I. CONCEPTION DÉTAILLÉE DES APPLICATIONS	
II. DÉVELOPPEMENT	
III. INTÉGRATION	
V. TESTS.....	
CHAPITRE 5 : AUTRES TÂCHES RÉALISÉES.....	
CONCLUSION	
BIBLIOGRAPHIE	3
SITES INTERNET UTILES	9
ANNEXES	

----- Saut de page -----

Liste des figures

Erreur ! Aucune entrée de table d'illustration n'a été trouvée.

----- Saut de page -----

Liste des annexes

ANNEXE A : La table des codes ASCII

ANNEXE B : Code source du

----- Saut de page -----

Glossaire

(Classé par ordre alphabétique ou d'apparition dans le texte, avec, éventuellement le numéro des pages)

M2 SIA :

Widget :

¹ Inutile dans le contexte de ce dossier de programmation.

² Ce n'est qu'un exemple de ce qu'on peut avoir à placer dans une table des matières

Introduction

Dans le cadre de mon Master **XXX** (diplôme) de l'Université Paul Sabatier de Toulouse, j'ai été amené à effectuer un stage du **XXX** au **XXX** au sein de l'entreprise **XXX**, implantée à **XXX** (Haute Garonne).

Ce stage s'est déroulé au sein de l'agence **XXX**, qui fait partie de la branche **XXX** de l'équipe **XXX**.

[Une introduction doit présenter :

- la problématique générale dans laquelle le travail réalisé va viser certains objectifs*
- les objectifs spécifiques du travail, dans le cadre de la problématique générale*
- le plan du rapport, de manière ni trop détaillée, ni trop laconique (une phrase par partie/chapitre n'est pas suffisant)*

]

La problématique abordée est

.....

Dans le cadre de cette problématique, l'objectif de ce stage est de **XXX**.

.....

Ces phrases NE SONT PAS A REPRENDRE
TELLES QUE ... MAIS A ADAPTER/COMPLÉTER

Dans ce rapport seront présentées les diverses étapes menant à la réalisation des tâches demandées. Les annexes présentes à la fin de ce document illustrent ce rapport et de nombreux points les référencent.

Après avoir présenté l'entreprise **XXX** et la branche **XXX**, on s'intéressera au projet **XXX**, à l'environnement de travail du stage, puis au sujet de stage dont la réalisation est présentée dans les deux chapitres suivants....

1. Objectifs généraux

a) Objectifs généraux du concept de "projet"

- 1) Traiter **toutes les étapes TECHNIQUES**, de **GESTION** et d'**ORGANISATION** d'un projet : **de l'analyse d'un cahier des charges** à la réalisation, en passant par une conception puis une programmation modulaire, l'intégration des divers composants conçus, **les tests et la validation**, tant par les concepteurs que par le client.
- 2) Traiter les **"AUTRES" TYPES D'ACTIVITÉS** liées à un projet :
 - **Restitution ÉCRITE** du travail technique : rédaction d'un rapport de travail et/ou d'un dossier d'analyse/conception et, dans le cas idéal, du manuel d'installation et du manuel d'utilisation du logiciel réalisé (ces 2 derniers rendus écrits sont optionnels dans ce projet).
 - **Restitution ORALE** du travail technique aux "clients" demandeurs du travail :
 - présentation en temps limité
 - réponses aux questions
 - démonstration du produit réalisé, avec réponse aux questions
 - **Perception des apports et des contraintes techniques et humaines du travail en équipe.**

b) Objectifs spécifiques/techniques du projet donné

- 1) Parfaire et compléter vos connaissances techniques sur la conception orientée objet et le langage C++, notamment, mais pas seulement, sur la manipulation de fichiers, de structures dynamiques, etc., en appliquant le tout à un sujet concret en rapport avec un des domaines de la formation.
- 2) Le langage C++ est potentiellement un outil pour la mise en œuvre de problèmes dans d'autres matières de la formation. Ce projet est donc une préparation et/ou un complément à d'autres matières de la formation, comme par exemple le traitement numérique des images (qui nécessitent de savoir ouvrir un fichier image, lire et traiter les informations qu'il contient, et sauvegarder dans un fichier les résultats des traitements).
- 3) Plus précisément, les traitements de fichiers image ou signal sont réalisés sous Matlab dans les autres UEs. Un des objectifs de ce projet est de mettre en œuvre dans un autre langage (C++) des algorithmes déjà vus et réalisés dans d'autres UEs. Cela devrait permettre de :
 - montrer le caractère générique d'un algorithme (hors considération de langage de mise en œuvre),
 - comparer les avantages et inconvénients de différentes mises en œuvre.

2. Évaluations du projet

Ce projet donnera lieu à une évaluation multifacette :

- 1) Un dossier d'analyse/conception sera à rendre au plus tard quelques jours avant la soutenance.
- 2) Une présentation orale, suivie de questions et d'une démonstration de l'application.

a) Le dossier d'analyse/conception

Le dossier se limitera à une vingtaine de pages au maximum par équipe, hors page de couverture, table des matières ... hors code source (à fournir en annexe), et hors détails des jeux de tests (à fournir en annexe).

- 1) **Structure du rapport** : page de garde, introduction (qui ? dans quelle formation ? objectifs ? plan), table des matières, ..., conclusion.

2) Contenu exigé :

- **Architecture globale de votre application et de ses principales fonctionnalités.**
- **Organisation de l'équipe et contribution de chaque membre : il faudra obligatoirement donner la répartition des tâches et qui a fait quoi dans l'équipe**
- Principales structures de données utilisées
- Des Algorithmes³ pour des parties délicates ou très particulières et nécessaire à la bonne compréhension (***ne pas confondre "algorithme" et "recopie de code source" ...***).
- Pour chaque classe, attribut et méthode : son rôle, et, le cas échéant, celui de ses paramètres et de la grandeur retournée et leur type. Etant précisé que :
 - Il n'est pas nécessaire de mettre dans le rapport tous les détails de toutes les classes, au besoin, utiliser des annexes
 - Certains détails peuvent n'apparaître que dans le code, en tant que commentaires
- **Jeux de tests utilisés pour la vérification** : avec explication de leur pertinence et résultats (par exemple, donner les "imprime-écran", ou rediriger les résultats dans un fichier à imprimer).
- Des remarques sur les points délicats d'analyse et de programmation.
- **Justification des principaux choix : tous les principaux choix devront être justifiés :**
 - outils (IDE, gestion de version, ect.), plateformes (Linux, Windows, ...), langages
 - choix de structuration de l'application
 - choix de votre organisation dans l'équipe
 - autres ...
- Si nécessaire, un petit guide d'utilisation, voire d'installation, pourra être mis en annexe également.
- Pour l'ensemble du projet, indiquer :
 - une estimation du temps consacré à ce projet (analyse, programmation, intégration et tests),
 - un aperçu des acquis et des difficultés rencontrées.

3) Modalités de remise (rendu) du rapport

Tout document sera à rendre par voie électronique, MAIS en respectant certaines règles, pour limiter le temps nécessaire pour enregistrer, renommer, ouvrir, imprimer et trier ... des centaines de fichiers.

- les noms des fichiers doivent être explicites ... pour identifier les auteurs et le contenu des fichiers.
- pas de format de fichier exotiques : uniquement du pdf, et en format "consultable" (recherche de texte possible) et non en format "image"
- on vous demande de rendre tout dans 1 seul fichier pour le rapport, les annexes et les codes. Si impossible, expliquer pourquoi c'est impossible.
- ATTENTION : veiller à la taille des fichiers, ... qui peut empêcher leur envoi par mail

b) La soutenance DEVRAIT être ... (avec N = nombre de personnes par équipe)

1) Chaque équipe de N étudiants dispose de :

- présentation : 15 minutes, à partager de manière équitable entre chaque membre de l'équipe (**aucun dépassement autorisé, et une sous-utilisation du temps sera également pénalisée**)
- questions : N fois 4 à 5 mn, mais ça peut durer plus longtemps
- démonstration (avec questions) : N fois 4 à 5 mn → à préparer : programme chargé sur un poste de travail, avec accès aux jeux de test
- commentaires (retours sur la "prestation") : 5 à 10 mn
- délibération : non précisé

2) Chaque membre de l'équipe DOIT PARTICIPER **ACTIVEMENT et INDIVIDUELLEMENT à chaque phase de la soutenance** : présentation orale, démonstration, et aux questions. **A VOUS DE VOUS ORGANISER POUR QUE CELA SOIT RESPECTÉ. LES MEMBRES DU JURY NE S'OCCUPERONT PAS DE FAIRE RESPECTER CETTE RÈGLE : ILS NE FERONT QU'APPRÉCIER LE RESPECT OU NON de CETTE RÈGLE. LE NON RESPECT DE CETTE RÈGLE INTERVIENDRA DANS LA NOTE COLLECTIVE DE L'ÉQUIPE ET LA NOTE INDIVIDUELLE DE SES MEMBRES.**

3) Tableau, vidéoprojecteur et PC (d'une salle de l'AIP) seront mis à disposition.

³ **Rappel** : un algorithme, c'est un glossaire des variables (nom, rôle, nature, plage de valeur, valeurs initiales), et le corps de l'algorithme qui ne doit utiliser que les concepts de séquences, choix et boucle

1. Cahier des charges

Il faut réaliser une application logicielle pour gérer des **ensembles** (nommées « bibliothèques » dans la suite de ce document) de fichiers signal/image et réaliser des traitements sur le contenu de ces fichiers.

a) Description générale

- ⇒ Chaque image est stockée dans un fichier. Il existe plusieurs types, ou format de fichier.
- ⇒ Une « image », est « composée » de deux « catégories » de données :
 1. un "contenu" : des données codant les éléments constitutifs (pixels) de l'image, et, selon le format du fichier, des données supplémentaires sur l'image (taux de compression, couleurs, ...),
 2. un « descripteur » ou « fiche technique » ou « lot de caractéristiques de l'image » : lot de données représentent les caractéristiques de l'image (titre, prix, auteurs, source, ...).

On peut faire l'analogie avec un catalogue ou un magasin : il y a le produit lui-même (analogie avec le point 1.) et à côté, une étiquette qui comporte un « descriptif textuel » (analogie avec le point 2.).

- ⇒ Dit autrement, dans ce projet, il faut considérer 2 catégories de données à manipuler :
 1. l'image elle-même (son contenu), stockée dans un fichier,
 2. une fiche « technique » ou « caractéristique » de l'image, externe au fichier image.

Par exemple, pour l'affichage d'informations sur une image de Mars, il s'agit d'afficher :

1. pour le point 1. : l'image elle-même,
2. pour le point 2. : seulement, par exemple : « titre : Mars ----- source : Nasa ----- numéro : 1 »



- ⇒ L'application à développer devra permettre à l'utilisateur de :
 1. **Gérer l'ensemble des « fiches techniques »** des images, et **afficher les images elles-mêmes, et générer quelques indicateurs à partir des « fiches techniques »** :
 - nombre d'images, coût (min, max, moyenne), nombre des images à accès libre, ...
 2. **Réaliser des traitements sur le contenu même des fichiers image** : filtrage par convolution,

Ces fonctionnalités peuvent être directement décrites dans le même langage de mise en œuvre que celui que vous utiliserez pour la gestion des descripteurs des images, mais aussi par appel à d'autres applications, comme Matlab ..., des applications spécifiques de traitement d'image, ...
 3. **Interagir avec l'application via une « IHM »**
 4. **Identifier les utilisateurs** : de plus, tous les utilisateurs ne doivent pas pouvoir accéder à n'importe quelle partie d'une bibliothèque. Il faut donc pouvoir **identifier** chaque utilisateur, pour déterminer à quelle partie il peut accéder et quelles fonctionnalités de l'application de gestion de la bibliothèque il peut utiliser.

b) Partie gestion des « fiches techniques » - l'application doit permettre de :

- ⇒ Construire la « bibliothèque d'images » (sous forme de liste de "descripteurs" d'images).
- ⇒ Gérer l'utilisation des différentes fonctionnalités de l'application, selon les droits de chaque utilisateur.
- ⇒ Afficher :
 1. intégralement la liste de descripteurs des images de la partie de la bibliothèque correspondant aux droits de l'utilisateur courant,
 2. le coût d'une image particulière, dont le numéro^(*) aura été saisi au clavier,
- ⇒ Mettre à jour cette bibliothèque (au niveau des descripteurs) :
 1. supprimer l'image dont l'utilisateur donne le numéro^(*),
 2. ajouter une image dont l'utilisateur donnera toutes les caractéristiques,
 3. modifier une ou des caractéristiques d'une image dont l'utilisateur donne le numéro^(*)

^(*) ou un autre paramètre au choix (à justifier) des concepteurs ; ces paramètres sont à proposer à l'utilisateur.

- ⇒ Trier la bibliothèque, suivant différents critères :
 1. UN critère **imposé** : ordre décroissant de coût,
 2. **Au moins deux autres critères choisis par les concepteurs du projet** : à définir et justifier
 3. Chaque résultat de tri doit pouvoir être affiché à l'écran
- ⇒ Construire **puis** afficher, à partir de la saisie au clavier d'une des caractéristiques des images, **une sous-liste des descripteurs des images correspondant à cette caractéristique**.
 1. UNE caractéristique est imposée : la plage de coût. L'utilisateur devra pouvoir afficher, à la demande, chacune des sous-listes correspondant à :

<i>gratuit</i>	<i>coût $\leq 9,99$ €</i>	<i>$10 \leq \text{coût}$ et $\text{coût} \leq 99,99$ €</i>	<i>coût > 100 €</i>	<i>des définitions plus fines : exemple, une plage coûts [min, max] à choisir par l'utilisateur</i>
----------------	--------------------------------------	--	-------------------------------------	---

2. **DEUX** autres caractéristiques seront **à définir par les concepteurs, en précisant pourquoi ils les ont jugées pertinentes à proposer comme critère de choix à un utilisateur**.
- ⇒ Sauvegarder la **bibliothèque** dans un fichier, et charger (récupérer), à partir d'un fichier, une bibliothèque existante. De plus, l'application doit permettre de sauvegarder et charger des noms différents de fichiers.
 - ⇒ Appliquer un traitement à l'image sélectionnée par l'utilisateur (voir ci-après).

c) Partie traitements d'une image - l'application doit permettre de :

L'objectif du projet n'est pas l'étude détaillée et la programmation massive de méthodes sophistiquées de traitement, mais seulement :

- 1) de montrer comment intégrer de telles méthodes dans une application plus générale,
- 2) de montrer le caractère générique (mise en œuvre dans différents langages) d'un algorithme de traitement d'image,
- 3) de comparer les avantages et inconvénients de différentes mises en œuvre.

Des exemples de traitement possibles, et très couramment utilisés :

- 1) calcul de l'histogramme d'une image,
- 2) filtrage d'une image par un filtre de convolution (matrice 3*3).
- 3) détection de contours avec filtres dérivatifs (gradient ou Laplacien)
- 4) détection de droites par la transformée de Hough (à faire après la détection de contours par exemple)
- 5) rehaussement de contours (amélioration de la netteté par rajout de contours détectés par Laplacien)
- 6) segmentation d'images par sélection de couleurs (intervalles de valeurs dans les trois composantes couleurs par exemple).

Le principe de ces traitements est normalement donné dans d'autres matières de la formation, et vous pouvez aussi faire des recherches sur ce sujet (sans y passer un temps excessif ...).

Les encadrants du projet ne sont pas à même d'expliquer les détails de ces algorithmes : en cas de questions spécifiques, renseignez-vous auprès des enseignants qui vous ont fourni ces algorithmes dans les autres matières.

Dans ce cadre :

- ⇒ CHAQUE MEMBRE de l'équipe DOIT METTRE EN ŒUVRE UN ALGORITHME PERSONNEL DIFFÉRENT, A CHOISIR DANS LA LISTE CI-DESSUS.
- ⇒ L'application doit être conçue pour pouvoir ajouter le plus facilement possible d'autres types de traitements que ceux qui seront déjà proposés (capacité « d'évolutivité » d'une application).
- ⇒ **La bibliothèque OPENCV devra être utilisée, au moins pour l'affichage des images initiales et traitées.**

d) Partie IHM

Cette IHM devra nécessairement être, au moins partiellement graphique. Le choix des technologies est libre.

2. Éléments de conception

a) Hypothèses (en vrac)

- ⇒ Le langage de mise en œuvre principale doit être le C++.
- ⇒ Pour les fichiers images, peut-être qu'un jeu d'images types sera fourni en cours de projet, mais ce n'est pas sûr. Donc, dans tous les cas, vous devez par vous-même choisir un jeu d'images à récupérer et à proposer au client. Des fichiers simples à manipuler sont par exemple ceux au format **.raw** (simple fichier texte, où chaque caractère représente un pixel codé sur 8 Bits ; ces fichiers ne contenant pas d'entête, il suffit donc pour les exploiter, de seulement connaître leur nombre de lignes et de colonnes). Autre exemple : les fichiers **.pgm**. Mais tout autre format jugé plus pertinent ou plus facile à manipuler (à justifier) pourra être utilisé. Se référer aux informations données dans d'autres matières.
- ⇒ Pour les droits d'accès à la bibliothèque, on distingue 3 catégories d'utilisateurs :
 1. Les utilisateurs de niveau 2, qui n'ont aucun droit.
 2. Les utilisateurs de niveau 1, dont les droits sont limités : accès en consultation et uniquement à une partie de la bibliothèque (partie définie par les utilisateurs de niveau 0)
 3. Les utilisateurs de niveau 0, qui ont tous les droits : accès à toutes les images, avec tous les droits (création/suppression, modification, consultation, traitement).

Un utilisateur doit s'identifier avec un code composé d'une suite de 2 lettres et 3 chiffres. Ce code permet de déterminer la catégorie de l'utilisateur qui se « connecte », et ainsi de connaître ses droits d'accès et de traitement sur la bibliothèque.

La connexion à l'application doit impérativement suivre le principe ci-dessus. Attention : vous ne devez pas concevoir et développer un vrai système de création et de gestion de comptes utilisateurs. Il s'agit uniquement d'effleurer la problématique de l'identification et de la distinction de différents utilisateurs
- ⇒ Le choix du type (texte ou binaire) des différentes catégories de fichiers nécessaires à la réalisation du projet est laissé au concepteur.
- ⇒ La bibliothèque d'images sera construite à l'aide d'une liste simplement chaînée, où toute nouvelle photo est ajoutée en FIN de liste.
- ⇒ Les fichiers contenant les descripteurs devront être de type texte.
- ⇒ Un numéro d'image ne pourra être attribué qu'à une seule image à un instant donné.

b) Structures de données

- ⇒ Une photo aura au moins les caractéristiques suivantes (mais pourra être complétée au besoin, toujours avec les justifications de ces compléments) :
 - **source** : chaîne de caractères formée d'un seul mot
 - **titre** : chaîne de caractères formée de plusieurs mots
 - **numéro** : un entier
 - **coût** : un réel
 - **accès** : un seul caractère (O : accès Ouvert à tous, R : accès Restreint aux utilisateurs de niveau 0)
 - **XXX** : une caractéristique à définir par les concepteurs, et qui servira à un utilisateur comme critère de sélection d'une sous-liste d'image.

3. Organisation et éléments évalués

- Vous devez vous organiser pour arriver à atteindre un maximum de objectifs du projet. Si vous n'avez pas le temps de tout faire, faites des choix judicieux mettant en valeur les principales connaissances/compétences que vous aurez acquises ⇒ bien analyser au préalable le problème pour le décomposer en sous-problèmes, identifier les "indépendances", se répartir le travail et choisir les parties à traiter en priorité.
- Il vaut mieux une application n'offrant qu'une partie des fonctionnalités demandées dans le cahier des charges, mais qui fonctionnent, plutôt qu'une application qui prévoit tout mais où rien ne marche !
- La priorité est au respect du cahier des charges : il ne faut pas partir dans des solutions et dans l'utilisation de technologies qui ne sont pas nécessaires, si c'est au détriment du respect du cahier des charges.
- C'est un travail d'équipe où chacun doit contribuer à la conception puis au développement d'une partie tout en ayant une bonne connaissance de ce qu'ont fait les autres membres de l'équipe.
- Le tableau ci-dessous indique les principaux aspects évalués (cette liste n'est pas exhaustive).
- Lors de la présentation et de la démonstration, considérez-vous en situation d'être vis-à-vis d'un client ou de votre commanditaire.

	Dossier d'analyse et de conception	Code	Soutenance Démonstration Questions
Organisation/Répartition/Cohésion/Contribution dans l'équipe : du travail, des réponses et des manipulations	✓		✓
Respect du cahier des charges - État d'avancement - Valorisation du travail réalisé et des résultats - Honnêteté	✓		✓
Algorithmes : présentation, niveau de détail, pertinence	✓		
Justification des choix - Analyses de points de particuliers (difficulté) et leurs justifications	✓		✓
Structures de données : présentations et justifications	✓		✓
Expression orale/écrite	✓		✓
Structuration (plan) de l'exposé écrit/oral	✓		✓
Timing - Manipulation des supports utilisés -Attitude			✓
Rigueur/synthèse	✓	✓	✓
Lisibilité du code : noms des variables, disposition du code, commentaires		✓	
Efficacité du code = choix adaptés des types et des opérateurs, pas de code "inutile"		✓	
Robustesse du code - Convivialité de l'interface avec l'utilisateur	✓	✓	✓
Connaissances techniques (du C++, et autres)	✓	✓	✓
Pertinence des jeux d'essais	✓		✓
Temps passé - Acquis - Intérêts	✓		✓

- Une attention toute particulière devra être portée à la « robustesse » du code, c'est à dire en particulier s'assurer de la validité des données saisies par l'utilisateur.
- Le code devra être "propre", bien commenté et ne générant aucun « warning » à la compilation.
- Ne jamais oublier d'interagir avec le client en cas de doute lors des différentes étapes du projet, et tout particulièrement lors de l'analyse du cahier des charges, et en cas de problèmes/retards.

Bibliographie (ou références bibliographiques)

Tanenbaum A., *Réseaux (3^{ème} édition)*. InterEditions, 1996

Sites Internet utiles

Format .raw : [https://fr.wikipedia.org/wiki/RAW_\(format_d'image\)](https://fr.wikipedia.org/wiki/RAW_(format_d'image)) (dernière consultation : le XX/XX/XXXX)