



Université Clermont Auvergne

École Universitaire de Physique etd'Ingénierie

Diplôme : Master Imagerie et Technologie pour la Médecine

(TechMed)

Tableau de bord de rapport

**Présenté par :** LEHARA Lyes

**Thème :** Bibliothèque de traitement d'images en C+

**Dirigé par :** Mme. PERY Emile

**Année universitaire :** 2024/2025.

* **Compte rendu de la réunion**

|  |  |
| --- | --- |
| **Date** | **31 mars 2024** |
| **Taches réaliser** | 1. Rédaction de cahier de charge. 2. Comprendre le travail à réaliser 3. Planification des taches de la première semaine. 4. Suivre les étapes de gestion du projet    * **Conception**      + Compréhension de cahier de charge      + Compréhension de la problématique      + Conception de projet      + Entrainement sur les outils à utiliser (Programmation en C++).    * **Développement**      + Implémentation des bibliothèque namespace V1.1, V1.1 et V2.0    * **Test et validation**      + Réalisation des tests **unitaires**    * **Documentation et déploiement**      + Rédaction de la documentation finale et le rapport |
| **Difficulté rencontré** |  |
| **Solution apporté** |  |

* **Etat d’avancement de la première semaine**

|  |  |
| --- | --- |
| **Date de début - date de fin** | **01 Avril – 06 Avril 2025** |
| **Taches réaliser** | 1. Lecture des cours de C++ 2. Faire de recherche sur visuel studio code 3. Configuration de l’éditeur pour exécuter le C++ 4. Réaliser les exercices de 1 à 5 5. Préparation du rapport (mise en forme) 6. Planification des taches de la deuxième semaine |
| **Difficulté rencontré** | 1. Problème d’exécution de C++ sous Windows 2. Difficulté trouver sur les pointeurs en C++ |
| **Solution apporté** | * Installation d’un outil MinGW qui permet de compiler et d’exécuter des ficher C/C++. |

|  |  |
| --- | --- |
| **Date de début - date de fin** | **07 Avril – 13 Avril 2025** |
| **Taches à réaliser** | 1. Flux d’exécution et gestion des erreurs 2. Comprendre la surcharge des fonctions 3. Bien comprendre les pointeurs |
| **Difficulté rencontré** | 1. Utilisation de try catch et la gestion des erreurs. 2. Utilisation std::cerr, assert et static\_assert |
| **Solution apporté** | 1. Travailler sur des exercices supplémentaires |

* **Etat d’avancement de la 2ème semaine**

|  |  |
| --- | --- |
| **Date de début - date de fin** | **07 Avril – 13 Avril 2025** |
| **Taches à réaliser** | 1. Flux d’exécution et gestion des erreurs 2. Comprendre la surcharge des fonctions 3. Bien comprendre les pointeurs |
| **Difficulté rencontré** | 1. Utilisation de try catch et la gestion des erreurs. 2. Utilisation std::cerr, assert et static\_assert |
| **Solution apporté** | 1. Travailler sur des exercices supplémentaires |

* **Etat d’avancement de la 3ème semaine**

|  |  |
| --- | --- |
| **Date de début - date de fin** | 1**4 Avril – 18 Avril 2025** |
| **Taches à réaliser** | 1. Comprendre la surcharge d’opérateurs, fonction virtuelle et patron de class 2. Comprendre le codage des images 3. Travailler sur les exercices 10, 11, 12 |
| **Difficulté rencontré** | 1. **Operateur () et <<** 2. Utilisation de **friend** 3. Patron class |
| **Solution apporté** | 1. Faire des recherches sur google 2. Utilisation de openAI pour bien comprendre |

* **Etat d’avancement de la 4ème semaine**

|  |  |
| --- | --- |
| **Date de début - date de fin** | **14 Avril – 18 Avril 2025** |
| **Taches à réaliser** | 1. Comprendre le travail à réaliser dans namespace v1.0 2. Travailler sur allocation des images et création des images 3. Création des images blanche |
| **Difficulté rencontré** | 1. Algorithme de sinusoïdale et damier |
| **Solution apporté** | 1. Inspirer des codes sur github 2. Utilisation de openAI pour bien comprendre |

* **Etat d’avancement de la 5ème semaine**

|  |  |
| --- | --- |
| **Date de début - date de fin** | **05 – 12 mai 2025** |
| **Taches à réaliser** | 1. Lecture et écriture d’un fichier. raw 2. La conversion d'image d'un type en un autre 3. La conversion d'une image en niveaux de gris en image en fausses 4. Couleurs à l'aide d'une LUT 5. Tester les différents LUT sur des plusieurs images 6. Premier pas sur namespace v1.1 |
| **Difficulté rencontré** | 1. Conversion images 16 bits en 8 bits 2. Convertir images RGB en images en niveaux de gris 3. Application du LUT si l’image en niveau de gris ou RGB |
| **Solution apporté** | 1. Faire des recherches sur les algorithmes à utiliser 2. Inspirer des codes sur github 3. Utilisation de openAI pour bien comprendre |

* **Etat d’avancement de la 6ème semaine**

|  |  |
| --- | --- |
| **Date de début - date de fin** | **12 – 16 mai 2025** |
| **Taches à réaliser** | 1. Création de class Image de type template (type générique) 2. Implémentation des fonctions utilisé dans namespace 1.0 en orienté objet (namespace1.1) 3. Creation class template ImageRGB qui hérite de la class Image 4. Instancier les deux class Image et ImageRGB dans main |
| **Difficulté rencontré** | 1. Comprendre la méthode static et la méthode de la class 2. La class imageRGB qui hérite de la class Image |
| **Solution apporté** | 1. Faire des recherches sur les algorithmes à utiliser 2. Inspirer des codes sur github 3. Refaire des exos sur des class abstract et des méthodes statics. 4. Utilisation de openAI pour bien comprendre |

* **Etat d’avancement de la 7ème semaine**

|  |  |
| --- | --- |
| **Date de début - date de fin** | **19 – 28 mai 2025** |
| **Taches à réaliser** | 1. Création les deux class génériques abstracts Processing 1 et 2 2. Comprendre l’addition de deux images |
| **Difficulté rencontré** | 1. Problème de séparation .hpp et .cpp dans des class template 2. Je suis bloqué dans le code de l’addition de deux images |
| **Solution apporté** | 1. Mettre dans même fichier namespace.hpp tout les namespace selon les différentes versions v1.0, v1.1 et v2.0 |