

Cahier Des Charges

Cahier des specifications techniques - Inclus

Ariel Roman - Jérémie Serre - Mathias Quillot

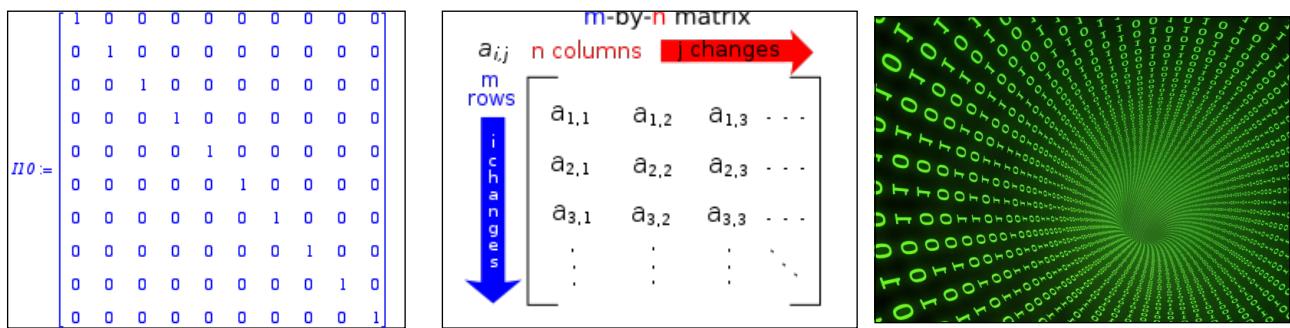
Licence Informatique Semestres 4

52768597	02605554864	22301123254	56452768597	02605554864	22301123254	56452768597	02605554864	2230112
97546567	52107905648	89780158595	45197546567	52107905648	89780158595	45197546567	52107905648	8978015
65666666	9201.265340	46243801255	67665666666	9201.265340	46243801255	67665666666	9201.265340	4624380
55468597	5326498235.	56897845022	66665468597	5326498235.	56897845022	66665468597	5326498235.	5689784
21342430	03125643754	24584686530	52421342430	03125643754	24584686530	52421342430	03125643754	2458468
29752834	34201326497	44565752389	43529752834	34201326497	44565752389	43529752834	34201326497	4456575
56749758	88260214687	70122648654	01356749758	88260214687	70122648654	01356749758	88260214687	7012264
01326798	95462032156	89901245984	53701326798	95462032156	89901245984	53701326798	95462032156	8990124
60546412	87546200012	56578021657	78765046412	87546200012	56578021657	78765046412	87546200012	5657802
01352679	56489854222	89535670000	56701352679	56489854222	89535670000	56701352679	56489854222	8953567
524.2134	30215021569	01444587901	886524.2134	30215021569	01444587901	886524.2134	30215021569	01444587
54240404	87459823654	89564875564	54654240404	87459823654	89564875564	54654240404	87459823654	89564875
21404359	85123030213	02654895465	23421404359	85123030213	02654895465	23421404359	85123030213	02654895
53402213	13311123150	13025165465	78553402213	13311000011	13025165465	78553402213	13311123644	13025165
58672464	25468952654	76540215497	49758672464	25468952654	76540215497	49758672464	25468952654	76540215
68652031	78021328503	87654860216	97968652031	78021328503	87654860216	97968652031	78021328503	87654860
79561203	57920045685	54897564202	25679561203	57920045685	54897564202	25679561203	57920045685	54897564
56530979	48314904153	15465465460	26456530979	48314904153	15465465460	26456530979	48314904153	15465465
32031246	18946516746	21654	SYSTEM FAILURE	18946516746	21654	18946516746	21654621	18946516746
56452123	51561687515	40216	SYSTEM FAILURE	51561687515	40216	51561687515	40216548	51561687515
45754545	23162685421	56102	SYSTEM FAILURE	45754545	23162685421	56102165	45754545	23162685421
91675425	62964975421	62165	SYSTEM FAILURE	62964975421	62165054	62964975421	62165054	62964975421
59782135	35656497652	13245450154	34659782135	35656497652	13245450154	34659782135	35656497652	13245450
23100002	31200124556	84987984301	64023100002	31200124556	84987984301	64023100002	31200124556	84987984
56462857	87976423120	24568765435	13656462857	87976423120	24568765435	13656462857	87976423120	24568765
15622256	31655976421	01235435435	55645622256	31655976421	01235435435	55645622256	31655976421	01235435
66566433	05234605242	43021648576	79866566433	05234605242	43021648576	79866566433	05234605242	43021648
23101346	59257561221	53441100000	59823101346	59257561221	53441100000	59823101346	59257561221	53441100
57242104	56024565237	00000001243	56457242104	56024565237	00000001243	56457242104	56024565237	00000001
68976543	85421245454	53727672034	23168976543	85421245454	53727672034	23168976543	85421245454	53727672
12124567	45456402124	25375763520	24212124567	45456402124	25375763520	24212124567	45456402124	25375763
12054976	24575454012	43597572672	54212054976	24575454012	43597572672	54212054976	24575454012	43597572
23051564	42245454440	40133727967	85323051564	42245454440	40133727967	85323051564	42245454440	40133727
46791630	55546520303	97801322479	65246791630	55546520303	97801322479	65246791630	55546520303	97801322
52675642	40555120245	69675014372	21352675642	40555120245	69675014372	21352675642	40555120245	69675014
21000231	21205512563	97846520434	13421000231	21205512563	97846520434	13421000231	21205512563	97846520
00000005	23564012452	52768975403	24000000005	23564012452	52768975403	24000000005	23564012452	52768975
04242112	54545450215	24214672732	42424242412	54545450215	24214672732	42424242412	54545450215	24214672
52424524	88879564501	03427679854	75452424524	88879564501	03427679854	75452424524	88879564501	03427679

257059237	88838289201	0370518388204	5204254392037	88838289201	0370518388204	5204254392037	88838289201	03705183
34343433	24242420312	34343434333	43434343433	24242420312	34343434333	43434343433	24242420312	34343433
00000002	33224073423	231689312403	34000000002	33224073423	231689312403	34000000002	33224073423	23168931
37000337	37302273283	33846230434	33451000337	37302273283	33846230434	33451000337	37302273283	33846230
23832043	40222730342	22632074935	37323612043	40222730342	22632074935	37323612043	40222730342	22632074
78387830	22246230303	23803333338	625362387830	22246230303	23803333338	625362387830	22246230303	23803333
33027282	43342424440	407332332881	62353027284	43342424440	407332332881	62353027284	43342424440	40733233
33026036	340212424072	402912130313	645373024035	340212424072	402912130313	645373024035	340212424072	40291213
33025936	340212424072	402912130313	645373024035	340212424072	402912130313	645373024035	340212424072	40291213

Présentation générale du projet

Le projet a pour but la réalisation d'une librairie que l'on va réaliser permettant la gestion de matrices creuses composées d'un nombre très élevé de lignes et de colonnes. Cette librairie est destinée à être utilisé par des Mathématiciens ou tout autre scientifique voulant faire des calculs complexes de matrice. Notre librairie permettra de faire des opérations arithmétiques de matrice (ex: addition, multiplication, soustraction, transposé, etc...)



Méthodologie et conception suivie

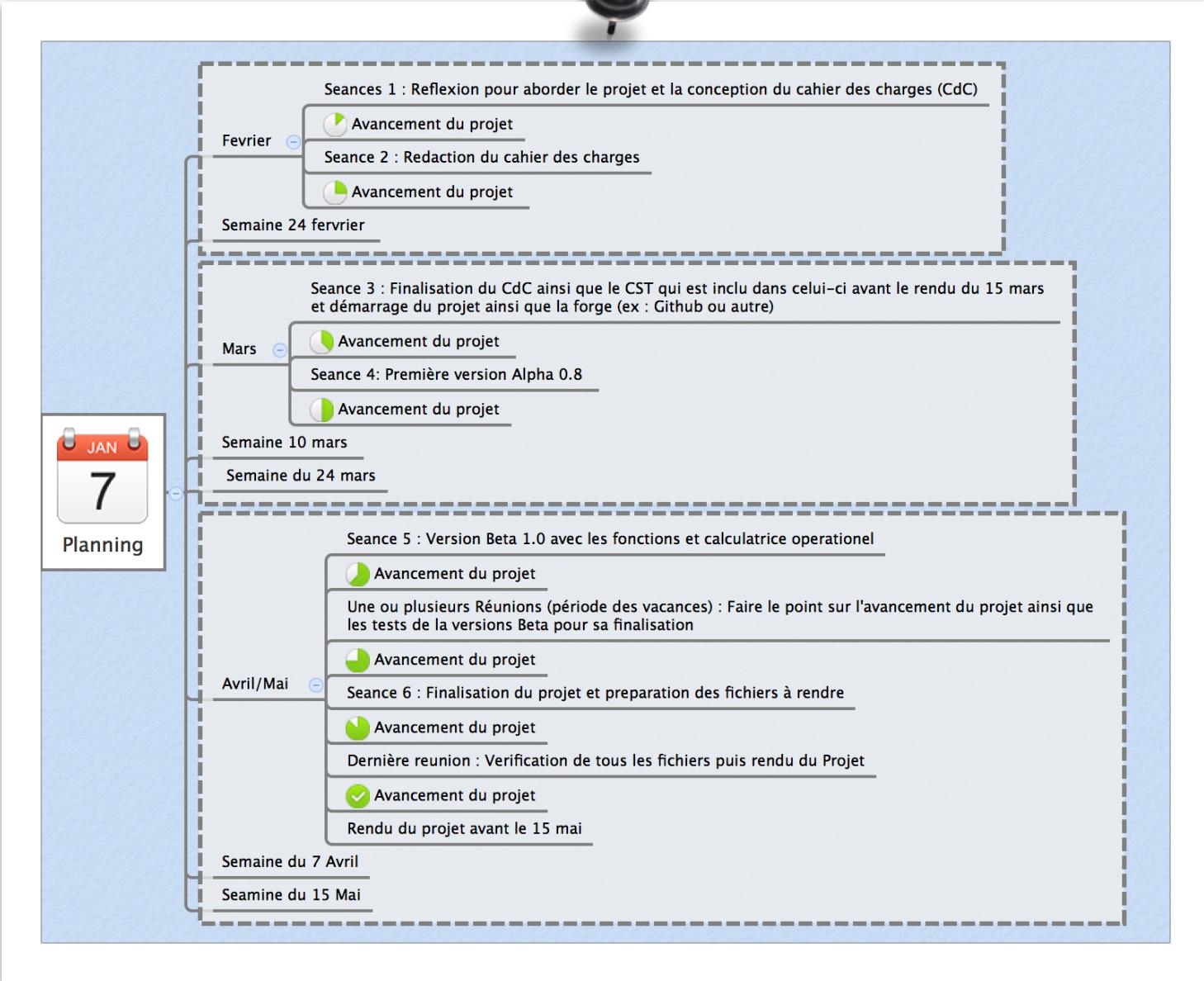
Notre méthodologie sera composée des étapes suivantes :

- **Planifier** : Approche et décomposition du projet suivi d'une planification.
- **Analyser** : Analyse approfondie du projet suivant les objectifs à atteindre.
- **Concevoir** : Modélisation de l'architecture du projet.
- **Construire** : Réalisation des différents modules suivis de leur assemblage au sein du projet.
- **Tester** : Phase de tests de performances et de fiabilité des données de sortie.
- **Déployer** : Rendu des fonctionnalités réalisées qui suivent les contraintes du CST.

Elle en sera précédé par une phase de configuration des outils nécessaires au suivi du projet : Forge, IDE, Machines.

Planning

Le planning suivra le schéma suivant :



Rendu du projet le 15 Mai au plus tard.

Les objectifs généraux de la librairie et leur critère de succès

Cette librairie a pour objectifs généraux de réaliser les opérations de base entre matrices tels que :

- l'addition
- La soustraction
- La multiplication
- La puissance
- La transposée
- L'inverse
- Et autre

Le critère de succès de ces fonctionnalités (opérations sur les matrices) dépendra de la fiabilité des résultats données par celles-ci dans les cas d'utilisation habituel. Aucune contrainte ne sera donné sur le temps d'exécution. Il sera fait au mieux pour que cette fiabilité s'opère aussi dans les cas critiques (surplus de données par exemple) et pour que ces opérations soient optimisées en temps et en dépense d'espace mémoire.

Les utilisateurs cibles et leurs besoins

Les personnes qui sont ciblées sont les suivantes :

- Les chercheurs
- Les informaticiens
- Les mathématiciens
- Les physiciens
- Les étudiants
- Toutes autres personnes ayant un lien avec la programmation.

Présentation des acteurs et leur responsabilités

Les acteurs seront les suivant :

- **Jérémy Serre** : Chef de projet qui a pour responsabilité l'organisation du projet, l'interaction avec le client ainsi que le rendu du projet.
- **Ariel Roman** : Développeur et testeur qui aura pour charge le codage des classes et le test de celles-ci.
- **Mathias Quillot** : Développeur et testeur qui aura pour responsabilité le codage des classes et le teste de celles-ci.

Nature des tests réalisés

- Vérification de la class matrice.
- Test des différentes fonctions de calcul.
- Test des fonctions de traduction (binaire -> texte / texte -> binaire).
- Test du bon fonctionnement du gestionnaire de matrice.
- Test du bon fonctionnement de la librairie sur programme annexe.
- Utilisation d'un Profiler pour optimiser le programme.

Cahier des Spécifications Techniques

Besoins fonctionnels

Les fonctions dont on a besoin sont les suivantes :

- Gestion d'une matrice creuse: Opérations dites plus tôt, accesseurs, mutateurs.
- Gestion du fichier associé à la matrice creuse: changer le nom d'un fichier, supprimer un fichier, créer un fichier, écriture dans le fichier. Ces fichiers sont en binaire. Conversion possible de fichier binaire à fichier texte et vice versa.
- Gestion de plusieurs matrices: Ajouter une matrice, supprimer une matrice, accès aux valeurs.

-
- Interprétation des expressions mathématiques: Interpréter des expressions qui en entrée (des lignes de commande à l'occurrence) et en générer une action en adéquation.

Structuration du projet

Cette librairie va être composées de différentes entités :

- MatriceCreuse: Matrice qui gère un fichier matrice creuse.
- MatricesManager : Un gestionnaire de plusieurs Matrices creuses et donc de plusieurs fichiers.
- InterpreteurMath : Un interpréteur qui récupérera des lignes de commandes et ordonnera à MatricesManager d'executer des actions.

Structuration de données

Une matrice creuse sera modélisée par la structure des données suivante :

[unsigned int] width [unsigned int] height
[unsigned int] Indice_i [unsigned int] Indice_j [int] value

Elle sera aussi convertible (de manière réversible) en un fichier texte ayant cette structure :

[string]width ; [string]height \n
[string]Indice_i ; [string]Indice_j ; [string]value \n // Chaque valeur différente de 0

Définitions des fonctions

A noter : Ces headers ne sont en aucun cas une finalité, ils servent juste de fil rouge au projet. Ils seront très certainement amenés à être modifier durant le process de développement.

- converter.h
- interpreter.h
- matrixManager.h
- sparseMatrix.h

```

// converter.h

#ifndef H_CONVERTISER
#define H_CONVERTISER

using namespace std;

class Converter
{
private:
    // Ici seront écrites les fonctions utilitaires
    // donc convert à besoin et dont l'utilisateur n'a pas le droit à l'accès.
public:
    static void convert(string &fileName1, string &fileName2);
};

#endif

```

```

// interpreter.h

#ifndef H_INTERPRETER
#define H_INTERPRETER

using namespace std;

class Interpreter
{
private:
    string command;
public:
    Interpreter();
    ~Interpreter();

    void activeProcess(MatrixManager & matrixManager); // Traite la commande
    //et active les fonctions adéquates au manageur de matrices.
    /* Exemple :
        Si la commande est C = A * B
        Il va faire matrixManager.multMatrix(matrixManager("A", "B", "C"));*/

    void setCommand(string &command); // Permet de donner une commande à l'interpréteur.
    string getCommand(); // Permet de récupérer la commande de l'interpréteur.
};


```

```

#ifndef H_MATRIXMANAGER
#define H_MATRIXMANAGER
#include <string>

//matrixManager.h

using namespace std;
class MatrixManager
{
private:
    SparseMatrix * matrix;

public:
    MatrixManager();
    ~MatrixManager();

    bool isSet(string & matrixA); // Vérifie que la matrice A existe.

    bool addMatrix(string &matrixA, string &matrixB, string &matrixOut); // On fait A+B et on met ça dans matrixOut.
    bool subMatrix(string &matrixA, string &matrixB, string &matrixOut); // On fait A-B et on met ça dans matrixOut.
    bool multMatrix(string &matrixA, string &matrixB, string &matrixOut); // On fait A*B et on met ça dans matrixOut.
    bool inverseMatrix(string &matrixA, string &matrixOut); // On inverse A et on met ça dans matrixOut.
    bool puishMatrix(string &matrixA, unsigned int n, string &matrixOut); // On fait A^n et on met ça dans matrixOut.
    bool transpMatrix(string &matrixA, string &matrixOut); // On fait la transposée de A et on met ça dans matrixOut.

    // A savoir que si MatrixOut n'existe pas, on la crée cette matrice.
    // A ajouter les mêmes fonctionnalités par rapport à des noms de fichier et non pas des noms de matrice.
    void setMatrix(ParseMatrix &matrixOut); // Crée une nouvelle matrice si jamais elle
                                                // n'existe pas.
    ParseMatrix getMatrix(string &matrix); // Récupère une matrice grâce à son nom.

};

#endif

```

```

// sparseMatrix.h
#ifndef H_SPARSEMATRIX
#define H_SPARSEMATRIX

#include <string>
using namespace std;
class SparseMatrix
{
private:
    string FileName;
    string name;
    unsigned int width;
    unsigned int height;

public:
    SparseMatrix();           // Mis à 0 de tout.
    SparseMatrix(string &fileName); // Elle fait référence à un fichier et en récupère ses champs.
    bool newMatrix(unsigned int &width, unsigned int &height, string &fileName); // Crée une nouvelle matrice.

    void random();
    string getFileName();      // On récupère le nom du fichier.
    void setFileName(string &fileName); // On modifie le nom du fichier et on récupère les nouvelles du champs s'il existe, sinon on réinitialise tout.

    void renameFile(string &fileName); // Renomme le fichier.

    // ACCESSEURS MUTATEUR
    unsigned int getWidth();        // Récupère la largeur.
    unsigned int getHeight();       // Modifie la largeur.

    string getName() const;
    void setName(string fileName);

};


```

};

~~void setFileName(string &fileName);~~

~~unsigned int getWidth();~~

Conventions nommage

Le code produit devra suivre les règles conventionnelles suivantes :

- **Anglais** : La seule langue autorisée sera l'anglais. Seuls les commentaires seront autorisées à être écrits dans une autre langue.
- **Les classes** : Commencent par une majuscule, et les mots sont séparés par une majuscule.

Par exemple :

- MaClasseDeBase
- CeciEstUneClasse

- **Les variables** : Commencent par une minuscule et les mots sont séparés par une majuscule. Si on a besoin de spécifier le type, on ajoute un '_' suivi du préfixe du type en minuscule. (choix du nombre de lettre contextuel).

Par exemple :

- maVariableDeClasse

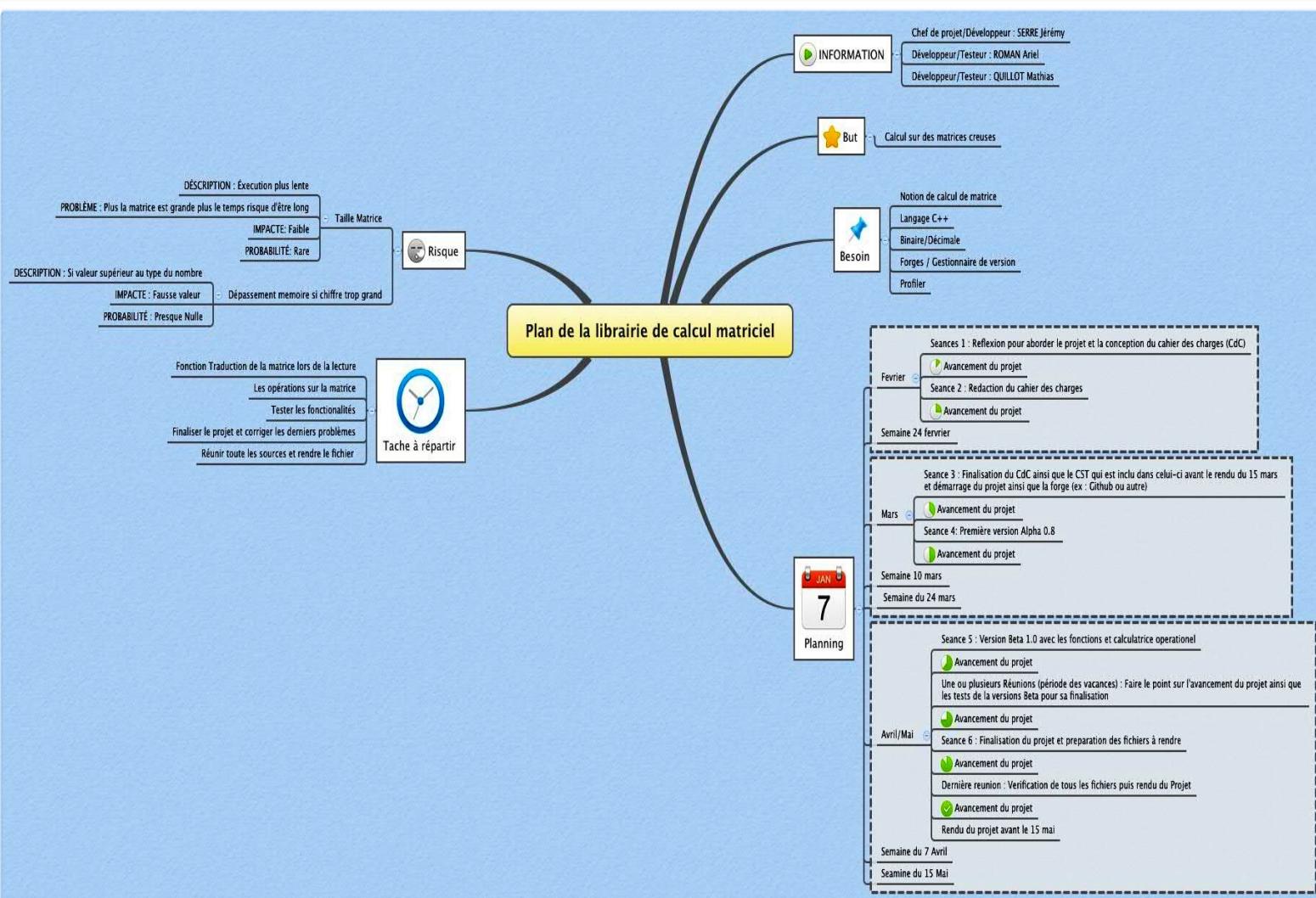
-
- maVariableEntiere_i (si risque de confusion avec une autre variable de même nom mais pas de même type)
 - variableDouble_do (s'il existe variableDouble_d pour la classe Debut par exemple)
-
- **Les fonctions/méthodes :** Commencent par une minuscule et les mots sont séparés par une majuscule. Les déclarations de fonctions doivent impérativement contenir le nom de leurs variables.
 - **Les accolades :** Les accolades sont mis à la ligne et sont suivies par un saut de ligne à part dans le cas d'un bloc vide.

Par exemple :

- MyClass
 - {
 - private:
 - int maVariable;
 - }
- void myFunction()
 - {
 - return 0;
 - }
- for(...)
 - {
- void myFunction(){}
 - {}

- **Les commentaires :** Suivre les conventions classiques de doxygen.
- **Les noms des fichiers :** Les noms des fichiers sont composés du nom de leur classe suivi de ".h" ou ".cpp" selon si c'est un header ou un code source. Si jamais il n'est pas associé à une classe, lui trouver un nom en liaison avec le module ou en rapport avec le code qui y est à l'intérieur.

Schéma de l'organisation du projet



Conctact

ADRESSE MAIL DE L'ÉQUIPE

Ariel Roman : ariel.roman@hotmail.fr

Jérémy Serre : serrejeremy@yahoo.fr

Mathias Quillot : mathiasquillot@gmail.com

ADRESSE POSTAL

Centre d'Enseignement et de Recherche en Informatique (C.E.R.I)

339 Chemin des Meinajaries,

04 90 XX XX XX

84140 Avignon

Plan

