|  |
| --- |
| Génération d'image fractale |

/\* image \*/

Auteur : HAUTIER Jonas

Email : [jonas.hautier@cpnv.ch](mailto:jonas.hautier@cpnv.ch)

Dernière modification : 06.05.2021

**Table des matières**

[1 Analyse préliminaire 3](#_Toc71184201)

[1.1 Introduction 3](#_Toc71184202)

[1.2 Objectifs 3](#_Toc71184203)

[1.3 Planification initiale 4](#_Toc71184204)

[2 Analyse / Conception 5](#_Toc71184205)

[2.1 Concept 5](#_Toc71184206)

[2.2 Stratégie de test 6](#_Toc71184207)

[2.2.1 Liste des tests / scénarios 6](#_Toc71184208)

[2.3 Risques techniques 8](#_Toc71184209)

[2.4 Planification 8](#_Toc71184210)

[2.5 Dossier de conception 10](#_Toc71184211)

[3 Réalisation 11](#_Toc71184212)

[3.1 Dossier de réalisation 11](#_Toc71184213)

[3.2 Description des tests effectués 11](#_Toc71184214)

[3.3 Erreurs restantes 11](#_Toc71184215)

[3.4 Liste des documents fournis 11](#_Toc71184216)

[4 Conclusions 12](#_Toc71184217)

[4.1 Objectifs 12](#_Toc71184218)

[4.2 Points positifs et négatifs 12](#_Toc71184219)

[4.3 Difficultés 12](#_Toc71184220)

[4.4 Suite pour le projet 12](#_Toc71184221)

[4.5 Comparaison entre la planification et le travail réalisé 12](#_Toc71184222)

[5 Glossaire 13](#_Toc71184223)

[6 Annexes 14](#_Toc71184224)

[6.1 Résumé du rapport du TPI 14](#_Toc71184225)

[6.2 Sources – Bibliographie 14](#_Toc71184226)

[6.3 Manuel d'installation 14](#_Toc71184227)

[6.4 Manuel d'utilisation 14](#_Toc71184228)

[6.5 Archives du projet 14](#_Toc71184229)

# Analyse préliminaire

## Introduction

L’objectif général de mon projet est d’implémenter un générateur d’image fractale.

Le programme sera développé en C#, utilisera SQLite comme système de gestion de base de données, sera versionné avec git et suivra la méthode de gestion de projet « Waterfall ».

Ce projet suivra les normes suivantes :

* La norme CPNV pour le MCD (Modèle Conceptuel de Données) et le MLD (Modèle Logique de Données) (accessible uniquement depuis la plateforme Moodle du CPNV)
* [Les normes de programmation et conventions de nommage en C#](https://github.com/ktaranov/naming-convention/blob/master/C%23%20Coding%20Standards%20and%20Naming%20Conventions.md) basées sur les recommandations de Microsoft

Ce projet est réalisé dans le condition du TPI, c.-à-d. :

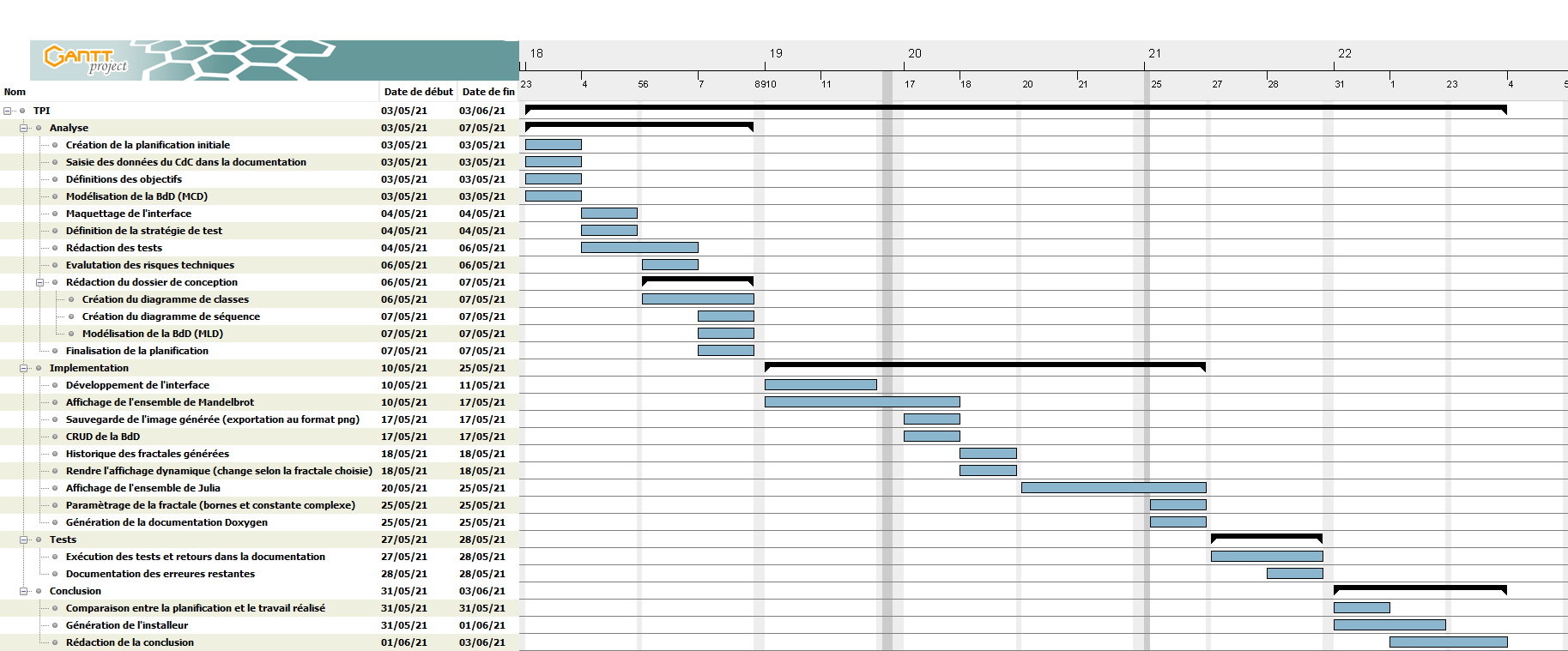
* Le projet est réalisé en 90h ;
* Le projet est individuel ;
* Le projet est commandé et évalué par le chef de projet ;
* Le rapport de projet est évalué par le chef de projet et deux experts ;
* La présentation et la défense sont évaluées par les experts.

Ce projet a pour but de confirmer que je suis capable d’exécuter un cahier des charges correspondant aux compétences d’un informaticien CFC.

## Objectifs

* Le programme est capable de générer l’une des fractales suivantes : l’ensemble de Mandelbrot et les ensembles de Julia de degré 2 ;
* Le programme donne à l’utilisateur la possibilité de définir les bornes réelles (x min et x max) et imaginaires (y min et y max) de la fractale générée ;
* Le programme adapte son affichage à la fractale choisie (c.-à-d., dans le cas des ensembles de Julia, il affiche un champ supplémentaire pour configurer *c*) ;
* Lorsque le programme a généré l’image de la fractale, il affiche le temps de génération de l’image ;
* Le programme donne à l’utilisateur la possibilité de sauvegarder l’image générée au format png ;
* Le programme est capable d’afficher un historique des derniers paramètres de génération utilisés ;
* Le programme donne à l’utilisateur la possibilité de régénérer une image à partir de l’historique ;

## Planification initiale



Ma planification initiale

# Analyse / Conception

## Concept

Voici les maquettes de mon programme :

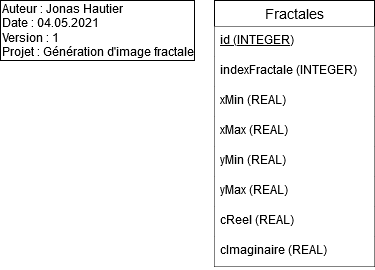
 

A gauche, la vue avec l’ensemble de Mandelbrot sélectionné.  
A droite, la vue avec l’ensemble de Julia sélectionné. Le paramétrage de « c » est visible.



La vue de l’historique

Ci-dessous, le MCD de mon programme :



MCD

## Stratégie de test

Pour mon projet, j’ai choisi d’effectuer la majorité de mes tests après avoir réalisé tous les modules de mon programme.

Après avoir développé un module, je ferai uniquement les tests unitaires puis passerai à la suite s’il fonctionne. Une fois les modules développés, je referai les tests unitaires, pour m’assurer de la non-régression de mes modules, puis ferai les tests d’intégration et d’acception, basés sur les objectifs du projet.

Tous les tests seront exécutés sur mon poste de développement. Par précaution, j’effectuerai les tests d’acceptation également sur une autre machine du CPNV pour m’assurer que le programme valide les objectifs.

Pour les données de tests, je générerai une dizaine de fractales et vérifierai qu’elles sont correctes avec l’application XaoS ([lien de téléchargement de XaoS](https://github.com/xaos-project/XaoS/)).

Les tests seront faits par moi-même et ne seront pas automatisés.

### Liste des scénarios

J’ai groupé mes scénarios selon les modules de mon programme.

#### Interface

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Etant donné | Lorsque | Alors |
| 1.1 | Le programme vient d’être lancé | Le programme est affiché à l’écran | La fractale sélectionnée est « Mandelbrot » |
| 1.2 | La fractale sélectionnée est « Mandelbrot » | Je sélectionne la fractale « Julia » | Les champs « c réel » et « c ima. » deviennent visibles |
| 1.3 | La fractale sélectionnée est « Julia » | Je sélectionne la fractale « Mandelbrot » | Les champs « c réel » et « c ima. » deviennent invisibles |
| 1.4 | Le programme vient d’être lancé | Le programme est affiché à l’écran | Les champs suivants contiennent :  x min 🡪 -2  x max 🡪 2  y min 🡪 -2  y max 🡪 2 |
| 1.5 | La limite minimale est plus grande que la limite maximal dans l’axe des réels | Je clique sur « Générer ! » | La fractale a subi une symétrie axiale, avec l’axe de symétrie sur l’axe vertical |
| 1.6 | La limite minimale est plus grande que la limite maximal dans l’axe des imaginaires | Je clique sur « Générer ! » | La fractale a subi une symétrie axiale, avec l’axe de symétrie sur l’axe horizontal |
| 1.7 | Une fractale a été générée | Le programme a fini de générer la fractale | Le temps de génération de la fractale est indiqué en bas à gauche de la fenêtre |

#### Génération de l’ensemble de Mandelbrot

Pour tous les tests de cette section, la fractale sélectionnée sera toujours « Mandelbrot ».

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Etant donné | Lorsque | Alors |
| 2.1 | Tous les champs sont remplis avec des données valides | Je clique sur « Générer ! » | La fractale est générée dans les limites données |
| 2.2 | Un des champs suivants n’a pas été rempli : « x min », « x max », « y min » ou « y max » | Je clique sur « Générer ! » | Le programme affiche une pop-up informant l’utilisateur du champ qu’il a omis |
| 2.3 | La fractale sélectionnée est « Mandelbrot » et un des champs suivants a été rempli avec des données invalides : « x min », « x max », « y min » ou « y max » | Je clique sur « Générer ! » | Le programme affiche une pop-up informant l’utilisateur du champ dans lequel une valeur incorrecte a été saisie |

#### Génération de l’ensemble de Julia

Pour tous les tests de cette section, la fractale sélectionnée sera toujours « Julia ».

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Etant donné | Lorsque | Alors |
| 3.1 | Tous les champs sont remplis avec des données valides | Je clique sur « Générer ! » | La fractale est générée dans les limites données |
| 3.2 | Un des champs suivants n’a pas été rempli : « x min », « x max », « y min », « y max », « c réel » ou « c ima. » | Je clique sur « Générer ! » | Le programme affiche une pop-up informant l’utilisateur du champ qu’il a omis |
| 3.3 | Un des champs suivants a été rempli avec des données invalides : « x min », « x max », « y min », « y max », « c réel » ou « c ima. » | Je clique sur « Générer ! » | Le programme affiche une pop-up informant l’utilisateur du champ dans lequel une valeur incorrecte a été saisie |

#### Historique des générations

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Etant donné | Lorsque | Alors |
| 4.1 | Le programme vient d’être installé et exécuté | Je regarde dans l’historique | L’historique est vide |
| 4.2 | Une fractale vient d’être générée | Je regarde dans l’historique | Les paramètres de génération de la fractale ont été ajouté à l’historique |
| 4.2.1 | Une fractale « Mandelbrot » vient d’être générée | Je regarde dans l’historique | Les paramètres de génération sont sous cette forme :  *Type : Mandelbrot x :* **xMin** *à* **xMax** *y :* **yMin** *à* **yMax** |
| 4.2.2 | Une fractale « Julia » vient d’être générée | Je regarde dans l’historique | Les paramètres de génération sont sous cette forme :  *Type : Julia x :* **xMin** *à* **xMax** *y :* **yMin** *à* **yMax** *c :* **cReel** *+/-* **cIma.** *i* |
| 4.3 | J’ai cliqué sur le bouton « Générer ! » mais il y avait un problème dans les paramètres de génération | Je regarde dans l’historique | L’historique contient les mêmes données/!\ |
| 4.4 | L’historique contient déjà des données | Je ferme le programme puis le relance | L’historique contient les mêmes données |
| 4.5 | L’historique est affiché et contient des données | Je clique sur l’une des entrées de l’historique | Le programme régénère la fractale sur laquelle j’ai cliqué |

#### Sauvegarde de la fractale générée

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Etant donné | Lorsque | Alors |
| 5.1 | Une fractale a été générée et est affichée dans la zone de dessin | Je clique sur la zone de dessin | Le programme ouvre une fenêtre pour enregistrer l’image générée |
| 5.2 | Une fractale a été générée et sauvegardée sur l’ordinateur | Je cherche le fichier sur mon ordinateur | Je trouve le fichier dans le répertoire défini à l’enregistrement et le nom de fichier est celui demandé |
| 5.3 | Une fractale a été générée et sauvegardée sur l’ordinateur | J’ouvre l’image générée | L’image correspond à la fractale générée |
| 5.4 | Le programme me propose d’enregistrer l’image de la fractale générée | Je saisi un nom de fichier invalide (dans un système NTFS) | La fenêtre « Enregistrer Sous » refuse le nom saisi et demande pour . |

/\* à mettre :

* Liste des tests par composant :   
  tableau contenant ID, Given, When, Then

\*/

## Risques techniques

/\* à mettre :

* Les risques techniques (complexité, compétences, …) et les solutions mises en place pour réduire les risques.

\*/

## Planification

/\* à mettre :

* Expliquer le choix de méthodologie (Agile/Waterfall)
* Planification définitive (en page suivante)

\*/

/\* Planification définitive\*/

## Dossier de conception

/\* à mettre :

* Le matériel hardware (justifier le choix)
* Les OS compatibles avec le programme (justifier le choix)
* Les logiciels utilisés, l’utilisation qui en est faite et la version utilisée
* MLD
* Diagramme de classe
* Diagramme de séquence

\*/

# Réalisation

## Dossier de réalisation

/\* à mettre :

* Structure du répertoire et liste des fichiers (avec description)
* Librairies utilisées (nom, auteur(s) et version)

\*/

## Description des tests effectués

|  |  |  |
| --- | --- | --- |
| Module | ID du test | Test passé ? |
| Interface du programme | 1.1 |  |
| 1.2 |  |
| 1.3 |  |
| 1.4 |  |
| 1.5 |  |
| 1.6 |  |
| 1.7 |  |
| Génération de l’ensemble de Mandelbrot | 2.1 |  |
| 2.2 |  |
| 2.3 |  |
| Génération de l’ensemble de Julia | 3.1 |  |
| 3.2 |  |
| 3.3 |  |
| Historique des générations | 4.1 |  |
| 4.2 |  |
| 4.2.1 |  |
| 4.2.2 |  |
| 4.3 |  |
| 4.4 |  |
| 4.5 |  |
| Sauvegarde de la fractale générée | 5.1 |  |
| 5.2 |  |
| 5.3 |  |
| 5.4 |  |

/\* à mettre :

* Liste des tests avec leur résultat

\*/

## Erreurs restantes

/\* à mettre :

* Lister les erreurs restantes
* Définir l’impact/les conséquences de l’erreur sur l’utilisation du produit
* Les actions à mettre en œuvre pour corriger le problème ou pallier à son impact négatif

\*/

## Liste des documents fournis

/\* à mettre :

* Les documents fournis au rendu (documentation, journal de travail, rapport du projet)

\*/

# Conclusions

/\* à mettre :

* Conclusion(s) générale(s) concernant le projet et sa réalisation

\*/

## Objectifs

/\* à mettre :

* Réflexion sur les objectifs atteints et, s’il y en a, les objectifs non-atteints

\*/

## Points positifs et négatifs

/\* à mettre :

* Réflexion sur les points positifs et négatifs du projet

\*/

## Difficultés

/\* à mettre :

* Réflexion sur les difficultés rencontrées, comment j’y ai réagi et comment j’aurais pu améliorer ma méthodologie
* Comparaison avec les risques techniques imaginés durant l’analyse

\*/

## Suite pour le projet

/\* à mettre :

* Les évolutions ou améliorations que je pense apporter au projet par la suite

\*/

## Comparaison entre la planification et le travail réalisé

/\* à mettre :

* Comparaison et réflexion sur les différences entre la planification et le travail réalisé

\*/

# Glossaire

# Annexes

/\* à mettre :

* Lister les documents annexes avec une description

\*/

## Résumé du rapport du TPI

/\* à mettre :

* 3 paragraphes : situation de départ, mise en œuvre et résultats
* Se référer aux critères d’évaluation pour plus de détails

\*/

## Sources – Bibliographie

Aide externes :

* 06.05.2020 – Louis Richard (camarade de classe) :  
  Louis m’a aidé pour détecter des cas qui n’auraient pas été pris en compte dans mes scénarios.

/\* à mettre :

* Les sources utilisées au format :
  + Pour les livres 🡪 titre, auteur, date
  + Pour les sites web 🡪 URL, date de consultation
  + Pour les articles 🡪 revue, titre, auteur, date
* Les aides externes au format : noms, raison, date

\*/

## Manuel d'installation

/\* à mettre :

* Comment installer le programme à partir de l’archive livrée

\*/

## Manuel d'utilisation

/\* à mettre :

* Comment utiliser les fonctionnalités du programme

\*/

## Archives du projet

/\* à mettre :

* Structure de l’archive
* Média utilisé pour livrer l’archive

\*/