

## Compte-rendu IN505

### 1. Bibliothèques nécessaires pour l'exécution du projet

Pour la réalisation du projet les bibliothèques Qt, OpenGL, Gl/glut, complex et cairo ont été utilisées et doivent être installées pour exécuter le programme

### 2. Mécanisme de compilation

Pour compiler, utiliser la commande make

Pour lancer le programme après compilation, utiliser la commande ./C-Project

le Makefile a été généré grâce à qmake avec les commandes suivantes :

qmake -project

puis rajout des lignes suivante dans C-project.pro :

QT += opengl

QT += widgets

CONFIG +=link\_pkgconfig

PKGCONFIG += gtk+-2.0

LIBS = -lglut -lGLU

puis

qmake

Pour exécuter le programme ./C-Project

### 3. Interface Qt

Pendant l'exécution du programme, les paramètres du dessin peuvent être changés grâce aux champs xmin, xmax, ymin, ymax et granularite. Pour appliquer les changements appuyez sur "ok". Les boutons "h", "b", "g", "d" permettent de se déplacer respectivement en haut, en bas, à gauche et à droite. Vous pouvez aussi changer de fractale avec les boutons "mandelbrot" et les 2 "julia" ayant comme complex  $c = -0.0519 + i0.688$  et  $c = -0.577 + i0.478$ . Le bouton "save" lance l'enregistrement de la fractale dessinée sous forme d'un fichier svg ayant pour nom "fractal.svg". Deux boutons de zoom/dezoom sont aussi présents. Appuyer sur la touche "echap" pour arrêter le programme.

## 4. Génération de la documentation

Pour créer un listing commenté et expliqué du programme, nous avons choisi d'utiliser Doxygen, pour cela nous avons commenté les .h du projet avec les conventions de Doxygen et avons choisi de sortir une documentation en html en créant et modifiant le fichier de configuration(Doxyfile) nécessaire à l'utilisation de Doxygen

commande utilisé :

```
doxygen -g //pour la création d'un fichier de configuration s'appelant Doxyfile
```

```
//puis modification du fichier de configuration
```

```
doxygen Doxyfile //pour la création de la documentation
```

Pour accéder à la documentation, allez dans le dossier html et ouvrez le fichier index.html dans un navigateur.

## 5. Etapes de la réalisation du projet

### a. Réflexion et recherche de documentation

Nous avons commencé par nous intéresser à l'utilisation des bibliothèques de dessin Cairo et Opengl ainsi que des algorithmes utilisés pour dessiner les fractales, nous avons aussi beaucoup discuté du sujet du projet pour s'assurer de notre compréhension de celui-ci. Nous avons aussi choisi de créer un dépôt distant sur Github pour avancer plus facilement à deux sur le projet

### b. Première utilisation des bibliothèques graphique

Après la recherche de documentation, nous avons réalisé deux petits programmes simple de dessin de la fractale de Mandelbrot avec cairo et opengl respectivement, nous avons ensuite décidé de dessiner la fractale dans un fichier svg avec cairo ce qui correspond pour nous au terme "d'enregistrement vectoriel" du sujet

### c. Réalisation de la classe abstraite Dessin

Suite à la réalisation des programmes de l'étape précédente, nous avons une bonne idée de comment implémenter la classe Dessin. Nous nous sommes donc attelé à cette tâche et avons adapté nos programmes à cette classe pour vérifier son bon fonctionnement. Enfin la classe Cairo a été améliorée pour s'approcher de sa version finale notamment grâce à la création de la classe point et de la réalisation de l'algorithme de dessin de la fractale de Julia et Fatou.

Cette étape nous a demandé beaucoup de discussion et de réflexion car l'implémentation de la classe dessin allait naturellement influer sur le fonctionnement de ses classes filles.

#### d. Utilisation de Qt avec openGL

Pour répondre au terme du sujet, nous devons utiliser openGL sous une interface Qt. Nous nous sommes donc renseigné sur la marche à suivre notamment grâce au lien figurant sur le sujet et avons ensuite implémenter cette fonctionnalité et avons modifié l'algorithme de dessin avec OpenGL en conséquence.

#### e. Relier les paramètre de dessin à l'interface Qt

Pour cette étape, une recherche sur les fonctionnalités de Qt a été nécessaire. Après plusieurs essais infructueux nous avons enfin réussi à implémenter des boutons permettant de modifier le cadre et la granularité du dessin.

#### f. Finition de l'interface Qt

Suite à notre compréhension de l'utilisation de Qt, nous avons ajouté les boutons d'enregistrement, d'actualisation du dessin, de zoom/dezoom, de changement de fractale (Mandelbrot à Julia et inversement) puis de déplacement.

#### g. Réalisation de la Documentation et dernière vérification

Le programme étant finis, il nous fallait maintenant écrire une documentation pour le projet, nous nous sommes rappelé de l'existence de Doxygen pour commenter le code et avons appris à créer une documentation avec celui-ci. les détails sont fournis page 1. Nous avons par ailleurs vérifié une dernière fois le code pour s'assurer d'aucune incohérence et de commentaires inutile non supprimés.