

Fractalium

1.0

1 Index des espaces de nommage	1
1.1 Liste des espaces de nommage	1
2 Index hiérarchique	3
2.1 Hiérarchie des classes	3
3 Index des classes	5
3.1 Liste des classes	5
4 Index des fichiers	7
4.1 Liste des fichiers	7
5 Documentation des espaces de nommage	9
5.1 Référence de l'espace de nommage Fractalium	9
5.1.1 Documentation des définitions de type	10
5.1.1.1 Double	10
5.1.2 Documentation des fonctions	10
5.1.2.1 serialize()	10
6 Documentation des classes	11
6.1 Référence de la structure Fractalium::Color	11
6.1.1 Documentation des données membres	11
6.1.1.1 b	11
6.1.1.2 g	11
6.1.1.3 r	12
6.2 Référence de la classe Fractalium::Complex	12
6.2.1 Documentation des constructeurs et destructeur	13
6.2.1.1 Complex() [1/2]	13
6.2.1.2 Complex() [2/2]	13
6.2.2 Documentation des fonctions membres	13
6.2.2.1 norm()	13
6.2.2.2 operator*()	13
6.2.2.3 operator+()	14
6.2.2.4 operator-()	14
6.2.2.5 operator/()	14
6.2.2.6 operator=()	15
6.2.3 Documentation des données membres	15
6.2.3.1 imag	15
6.2.3.2 real	15
6.3 Référence de la classe Fractalium::Fractal	15
6.3.1 Documentation des énumérations membres	16
6.3.1.1 FractalType	16
6.3.2 Documentation des constructeurs et destructeur	16
6.3.2.1 Fractal()	16

6.3.3 Documentation des fonctions membres	17
6.3.3.1 getType()	17
6.3.3.2 pointCheck()	17
6.3.3.3 setType()	17
6.3.4 Documentation des fonctions amies et associées	17
6.3.4.1 boost::serialization::access	17
6.4 Référence de la classe Fractalium::FractalWidget	18
6.4.1 Documentation des constructeurs et destructeur	18
6.4.1.1 FractalWidget()	18
6.4.1.2 ~FractalWidget()	19
6.4.2 Documentation des fonctions membres	19
6.4.2.1 disableSelection()	19
6.4.2.2 enableSelection()	19
6.4.2.3 mouseMoveEvent()	19
6.4.2.4 mousePressEvent()	19
6.4.2.5 mouseReleaseEvent()	20
6.4.2.6 newSelection	20
6.4.2.7 setFractal()	20
6.5 Référence de la structure Fractalium::History	20
6.5.1 Documentation des fonctions membres	21
6.5.1.1 serialize()	21
6.5.2 Documentation des données membres	21
6.5.2.1 image	21
6.5.2.2 offset	21
6.5.2.3 step_coord	21
6.6 Référence de la structure Fractalium::HistoryImage	22
6.6.1 Documentation des constructeurs et destructeur	22
6.6.1.1 HistoryImage() [1/2]	22
6.6.1.2 HistoryImage() [2/2]	22
6.6.2 Documentation des fonctions membres	22
6.6.2.1 serialize()	22
6.6.2.2 toQImage()	23
6.6.3 Documentation des données membres	23
6.6.3.1 image	23
6.7 Référence de la classe Fractalium::Image	23
6.7.1 Documentation des constructeurs et destructeur	24
6.7.1.1 Image() [1/3]	24
6.7.1.2 Image() [2/3]	24
6.7.1.3 Image() [3/3]	24
6.7.2 Documentation des fonctions membres	25
6.7.2.1 getPixel()	25
6.7.2.2 height()	25

6.7.2.3 merge()	25
6.7.2.4 operator=()	25
6.7.2.5 setPixel()	26
6.7.2.6 width()	26
6.7.3 Documentation des fonctions amies et associées	26
6.7.3.1 boost::serialization::access	26
6.7.3.2 operator<<	26
6.8 Référence de la classe MainWindow	27
6.8.1 Documentation des constructeurs et destructeur	27
6.8.1.1 MainWindow()	27
6.8.1.2 ~MainWindow()	27
6.8.2 Documentation des fonctions membres	27
6.8.2.1 event()	27
6.8.3 Documentation des données membres	28
6.8.3.1 instance	28
6.8.3.2 TOTAL_COLORS	28
6.9 Référence de la classe Fractalium::MPICalculator	28
6.9.1 Documentation des constructeurs et destructeur	29
6.9.1.1 MPICalculator()	29
6.9.2 Documentation des fonctions membres	30
6.9.2.1 calculate()	30
6.9.2.2 receive()	30
6.9.2.3 send()	30
6.9.2.4 stop()	31
6.9.3 Documentation des données membres	31
6.9.3.1 finshed	31
6.9.3.2 is_running	31
6.9.3.3 mpi_struct	31
6.9.3.4 node_count	31
6.9.3.5 node_recived	31
6.9.3.6 node_working	31
6.9.3.7 rank	32
6.10 Référence de la classe Fractalium::MPIStruct	32
6.10.1 Documentation des constructeurs et destructeur	33
6.10.1.1 MPIStruct() [1/3]	33
6.10.1.2 MPIStruct() [2/3]	33
6.10.1.3 MPIStruct() [3/3]	33
6.10.2 Documentation des fonctions amies et associées	33
6.10.2.1 boost::serialization::access	33
6.10.3 Documentation des données membres	33
6.10.3.1 end_x	34
6.10.3.2 end_y	34

6.10.3.3 fractal	34
6.10.3.4 height	34
6.10.3.5 iterations	34
6.10.3.6 offset_x	34
6.10.3.7 offset_y	34
6.10.3.8 start_x	34
6.10.3.9 start_y	35
6.10.3.10 step_coord	35
6.10.3.11 width	35
6.11 Référence de la classe PaintFractalEvent	35
6.11.1 Documentation des constructeurs et destructeur	35
6.11.1.1 PaintFractalEvent()	35
6.11.1.2 ~PaintFractalEvent()	36
6.11.2 Documentation des données membres	36
6.11.2.1 PaintFractalEventType	36
6.12 Référence de la structure Fractalium::Settings	36
6.12.1 Documentation des énumérations membres	37
6.12.1.1 CalculationType	37
6.12.2 Documentation des fonctions membres	37
6.12.2.1 init()	37
6.12.2.2 resetCrash()	38
6.12.2.3 saveSettings()	38
6.12.2.4 setCrash()	38
6.12.3 Documentation des données membres	38
6.12.3.1 AUTO_SNAPSHOTS	38
6.12.3.2 CALCULATION_TYPE	38
6.12.3.3 CRASH_SNAP_PATH	38
6.12.3.4 DISPLAY_SIZE_HEIGHT	39
6.12.3.5 DISPLAY_SIZE_WIDTH	39
6.12.3.6 GITHUB_URL	39
6.12.3.7 IS_CRASHED	39
6.12.3.8 ITERATIONS	39
6.12.3.9 NODES	39
6.12.3.10 SAVE_PATH	39
6.13 Référence de la classe SettingsDialog	40
6.13.1 Documentation des constructeurs et destructeur	40
6.13.1.1 SettingsDialog()	40
6.14 Référence de la classe SnapshotDialog	40
6.14.1 Documentation des énumérations membres	41
6.14.1.1 Return	41
6.14.2 Documentation des constructeurs et destructeur	41
6.14.2.1 SnapshotDialog()	41

6.14.3 Documentation des fonctions membres	41
6.14.3.1 returnType()	42
6.15 Référence de la structure Fractalium::SnapshotHistory	42
6.15.1 Documentation des constructeurs et destructeur	42
6.15.1.1 SnapshotHistory() [1/2]	42
6.15.1.2 SnapshotHistory() [2/2]	42
6.15.2 Documentation des fonctions membres	43
6.15.2.1 serialize()	43
6.15.3 Documentation des données membres	43
6.15.3.1 fractal	43
6.15.3.2 history_back	43
6.15.3.3 history_front	43
7 Documentation des fichiers	45
7.1 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Color.hpp	45
7.2 Color.hpp	45
7.3 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Complex.cpp	45
7.4 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Complex.hpp	46
7.5 Complex.hpp	46
7.6 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Double.hpp	46
7.7 Double.hpp	47
7.8 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Fractal.cpp	47
7.9 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Fractal.hpp	47
7.10 Fractal.hpp	48
7.11 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/FractalWidget.cpp	49
7.12 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/FractalWidget.hpp	49
7.13 FractalWidget.hpp	49
7.14 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/History.hpp	50
7.15 History.hpp	51
7.16 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Image.cpp	52
7.17 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Image.hpp	52
7.18 Image.hpp	53
7.19 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/mainwindow.cpp	53
7.19.1 Documentation des fonctions	54
7.19.1.1 getColorForDivergence()	54
7.19.2 Documentation des variables	55
7.19.2.1 c	55
7.19.2.2 color_mode	55
7.19.2.3 get_color	55
7.20 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/mainwindow.hpp	55
7.21 mainwindow.hpp	56
7.22 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/MPI.cpp	57

7.23 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/MPI.hpp	57
7.24 MPI.hpp	58
7.25 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Settings.cpp	59
7.26 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/Settings.hpp	59
7.27 Settings.hpp	60
7.28 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/SettingsDialog.cpp . . .	60
7.29 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/SettingsDialog.hpp . . .	61
7.30 SettingsDialog.hpp	61
7.31 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/SnapshotDialog.cpp . .	62
7.32 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/SnapshotDialog.hpp . .	62
7.33 SnapshotDialog.hpp	62

Chapitre 1

Index des espaces de nommage

1.1 Liste des espaces de nommage

Liste de tous les espaces de nommage avec une brève description:

Fractalium	9
--------------------------------------	---

Chapitre 2

Index hiérarchique

2.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Fractalium::Color	11
Fractalium::Complex	12
Fractalium::Fractal	15
Fractalium::History	20
Fractalium::HistoryImage	22
Fractalium::Image	23
Fractalium::MPICalculator	28
Fractalium::MPIStruct	32
QDialog	
SettingsDialog	40
SnapshotDialog	40
QEvent	
PaintFractalEvent	35
QLabel	
Fractalium::FractalWidget	18
QMainWindow	
MainWindow	27
Fractalium::Settings	36
Fractalium::SnapshotHistory	42

Chapitre 3

Index des classes

3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

Fractalium::Color	11
Fractalium::Complex	12
Fractalium::Fractal	15
Fractalium::FractalWidget	18
Fractalium::History	20
Fractalium::HistoryImage	22
Fractalium::Image	23
MainWindow	27
Fractalium::MPICalculator	28
Fractalium::MPIStruct	32
PaintFractalEvent	35
Fractalium::Settings	36
SettingsDialog	40
SnapshotDialog	40
Fractalium::SnapshotHistory	42

Chapitre 4

Index des fichiers

4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

/Users/maximecolliat/Documents/Projet/Fractalium/src/Color.hpp	45
/Users/maximecolliat/Documents/Projet/Fractalium/src/Complex.cpp	45
/Users/maximecolliat/Documents/Projet/Fractalium/src/Complex.hpp	46
/Users/maximecolliat/Documents/Projet/Fractalium/src/Double.hpp	46
/Users/maximecolliat/Documents/Projet/Fractalium/src/Fractal.cpp	47
/Users/maximecolliat/Documents/Projet/Fractalium/src/Fractal.hpp	47
/Users/maximecolliat/Documents/Projet/Fractalium/src/FractalWidget.cpp	49
/Users/maximecolliat/Documents/Projet/Fractalium/src/FractalWidget.hpp	49
/Users/maximecolliat/Documents/Projet/Fractalium/src/History.hpp	50
/Users/maximecolliat/Documents/Projet/Fractalium/src/Image.cpp	52
/Users/maximecolliat/Documents/Projet/Fractalium/src/Image.hpp	52
/Users/maximecolliat/Documents/Projet/Fractalium/src/mainwindow.cpp	53
/Users/maximecolliat/Documents/Projet/Fractalium/src/mainwindow.hpp	55
/Users/maximecolliat/Documents/Projet/Fractalium/src/MPI.cpp	57
/Users/maximecolliat/Documents/Projet/Fractalium/src/MPI.hpp	57
/Users/maximecolliat/Documents/Projet/Fractalium/src/Settings.cpp	59
/Users/maximecolliat/Documents/Projet/Fractalium/src/Settings.hpp	59
/Users/maximecolliat/Documents/Projet/Fractalium/src/SettingsDialog.cpp	60
/Users/maximecolliat/Documents/Projet/Fractalium/src/SettingsDialog.hpp	61
/Users/maximecolliat/Documents/Projet/Fractalium/src/SnapshotDialog.cpp	62
/Users/maximecolliat/Documents/Projet/Fractalium/src/SnapshotDialog.hpp	62

Chapitre 5

Documentation des espaces de nommage

5.1 Référence de l'espace de nommage Fractalium

Classes

- struct [Color](#)
- class [Complex](#)
- class [Fractal](#)
- class [FractalWidget](#)
- struct [History](#)
- struct [HistoryImage](#)
- class [Image](#)
- class [MPICalculator](#)
- class [MPIStruct](#)
- struct [Settings](#)
- struct [SnapshotHistory](#)

Définitions de type

- typedef boost::multiprecision::cpp_dec_float_100 [Double](#)

Fonctions

- template<typename Archive >
void [serialize](#) (Archive &ar, [Color](#) &color, const unsigned int version)

5.1.1 Documentation des définitions de type

5.1.1.1 Double

```
typedef boost::multiprecision::cpp_dec_float_100 Fractalium::Double
```

5.1.2 Documentation des fonctions

5.1.2.1 serialize()

```
template<typename Archive >  
void Fractalium::serialize (  
    Archive & ar,  
    Color & color,  
    const unsigned int version )
```

Chapitre 6

Documentation des classes

6.1 Référence de la structure Fractalium::Color

```
#include <Color.hpp>
```

Attributs publics

- int `r`
- int `g`
- int `b`

6.1.1 Documentation des données membres

6.1.1.1 `b`

```
int Fractalium::Color::b
```

6.1.1.2 `g`

```
int Fractalium::Color::g
```

6.1.1.3 r

```
int Fractalium::Color::r
```

La documentation de cette structure a été générée à partir du fichier suivant :

— [/Users/maximecolliat/Documents/Projet/Fractalium/src/Color.hpp](#)

6.2 Référence de la classe Fractalium::Complex

```
#include <Complex.hpp>
```

Fonctions membres publiques

— [Complex](#) ([Double](#) r, [Double](#) i)

Constructeur par défaut.

— [Complex](#) ()

Constructeur par défaut sans paramètres.

— [Complex](#) & [operator=](#) ([Complex](#) const &obj)

Retourne le carré du module du nombre complexe.

— [Complex](#) [operator+](#) ([Complex](#) const &obj) const

Surcharge de l'opérateur d'addition.

— [Complex](#) [operator-](#) ([Complex](#) const &obj) const

Surcharge de l'opérateur de soustraction.

— [Complex](#) [operator*](#) ([Complex](#) const &obj) const

Surcharge de l'opérateur de multiplication.

— [Complex](#) [operator/](#) ([Complex](#) const &obj) const

Surcharge de l'opérateur de division.

— [Double](#) [norm](#) () const

Retourne le module du nombre complexe.

Attributs publics

— [Double](#) [real](#)

— [Double](#) [imag](#)

6.2.1 Documentation des constructeurs et destructeur

6.2.1.1 Complex() [1/2]

```
Complex::Complex (
    Double r,
    Double i )
```

Constructeur par défaut.

Paramètres

<i>r</i>	partie réelle du nombre complexe
<i>i</i>	partie imaginaire du nombre complexe

6.2.1.2 Complex() [2/2]

```
Complex::Complex ( )
```

Constructeur par default sans paramètres.

6.2.2 Documentation des fonctions membres

6.2.2.1 norm()

```
Double Complex::norm ( ) const
```

Retourne le module du nombre complexe.

Renvoie

Double

6.2.2.2 operator*()

```
Complex Complex::operator* (
    Complex const & obj ) const
```

Surcharge de l'opérateur de multiplication.

Paramètres

<i>obj</i>	Nombre complexe à multiplier
------------	------------------------------

Renvoie[Complex](#)**6.2.2.3 operator+()**

```
Complex Complex::operator+ (
    Complex const & obj ) const
```

Surcharge de l'opérateur d'addition.

Paramètres

<i>obj</i>	Nombre complexe à ajouter
------------	---------------------------

Renvoie[Complex](#)**6.2.2.4 operator-()**

```
Complex Complex::operator- (
    Complex const & obj ) const
```

Surcharge de l'opérateur de soustraction.

Paramètres

<i>obj</i>	Nombre complexe à soustraire
------------	------------------------------

Renvoie[Complex](#)**6.2.2.5 operator/()**

```
Complex Complex::operator/ (
    Complex const & obj ) const
```

Surcharge de l'opérateur de division.

Paramètres

<i>obj</i>	Nombre complexe à diviser
------------	---------------------------

Renvoie

[Complex](#)**6.2.2.6 `operator=()`**

```
Complex & Complex::operator= (
    Complex const & obj )
```

Retourne le carré du module du nombre complexe.

Renvoie

[Complex](#)**6.2.3 Documentation des données membres****6.2.3.1 `imag`**

```
Double Fractalium::Complex::imag
```

6.2.3.2 `real`

```
Double Fractalium::Complex::real
```

La documentation de cette classe a été générée à partir des fichiers suivants :

- `/Users/maximecolliat/Documents/Projet/Fractalium/src/Complex.hpp`
- `/Users/maximecolliat/Documents/Projet/Fractalium/src/Complex.cpp`

6.3 Référence de la classe `Fractalium::Fractal`

```
#include <Fractal.hpp>
```

Types publics

— enum class `FractalType` {
 `Mandelbrot` , `Julia` , `BurningShip` , `Newton` ,
 `Koch` }

Fonctions membres publiques

— `Fractal` (`FractalType` type=`FractalType::Mandelbrot`)
 — int `pointCheck` (const `Complex` &pointm, int iterations) const

Fonction pour calculer le nombre d'itérations pour un point donné dans un ensemble de fractales.

— void `setType` (`FractalType` type)
 — `FractalType` `getType` () const

Amis

— class `boost::serialization::access`

6.3.1 Documentation des énumérations membres

6.3.1.1 FractalType

```
enum class Fractalium::Fractal::FractalType [strong]
```

Valeurs énumérées

Mandelbrot	
Julia	
BurningShip	
Newton	
Koch	

6.3.2 Documentation des constructeurs et destructeur

6.3.2.1 Fractal()

```
Fractalium::Fractal::Fractal (  

    FractalType type = FractalType::Mandelbrot ) [inline], [explicit]
```


6.3.3 Documentation des fonctions membres

6.3.3.1 getType()

```
FractalType Fractalium::Fractal::getType ( ) const [inline]
```

6.3.3.2 pointCheck()

```
int Fractal::pointCheck (
    const Complex & point,
    int iterations ) const
```

Fonction pour calculer le nombre d'itérations pour un point donné dans un ensemble de fractales.

Paramètres

<i>point</i>	
<i>iterations</i>	

Renvoie

6.3.3.3 setType()

```
void Fractalium::Fractal::setType (
    FractalType type ) [inline]
```

6.3.4 Documentation des fonctions amies et associées

6.3.4.1 boost::serialization::access

```
friend class boost::serialization::access [friend]
```

La documentation de cette classe a été générée à partir des fichiers suivants :

- /Users/maximecolliat/Documents/Projet/Fractalium/src/[Fractal.hpp](#)
 - /Users/maximecolliat/Documents/Projet/Fractalium/src/[Fractal.cpp](#)
-

6.4 Référence de la classe Fractalium::FractalWidget

```
#include <FractalWidget.hpp>
```

Graphe d'héritage de Fractalium::FractalWidget:

Graphe de collaboration de Fractalium::FractalWidget:

Signaux

- void [newSelection](#) (const QPoint &, const QPoint &)

Fonctions membres publiques

- [FractalWidget](#) (QWidget *parent=nullptr)

Constructeur de la classe [FractalWidget](#).

- [~FractalWidget](#) () override=default

- void [setFractal](#) (const QImage &image)

Permet de définir l'image de la fractale à afficher.

- void [disableSelection](#) ()

- void [enableSelection](#) ()

Fonctions membres protégées

- void [mousePressEvent](#) (QMouseEvent *event) override

Evènement de pression de la souris, permet de définir la zone de sélection de la fractale à afficher.

- void [mouseReleaseEvent](#) (QMouseEvent *event) override

Permet de définir la zone de sélection pour zoomer dans la fractale.

- void [mouseMoveEvent](#) (QMouseEvent *event) override

Dessine le rectangle de sélection sur la fractale.

6.4.1 Documentation des constructeurs et destructeur

6.4.1.1 FractalWidget()

```
FractalWidget::FractalWidget (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur de la classe [FractalWidget](#).

Paramètres

<i>parent</i>	Le widget parent
---------------	------------------

6.4.1.2 ~FractalWidget()

```
Fractalium::FractalWidget::~~FractalWidget ( ) [override], [default]
```

6.4.2 Documentation des fonctions membres**6.4.2.1 disableSelection()**

```
void Fractalium::FractalWidget::disableSelection ( ) [inline]
```

6.4.2.2 enableSelection()

```
void Fractalium::FractalWidget::enableSelection ( ) [inline]
```

Voici le graphe des appelants de cette fonction :

6.4.2.3 mouseMoveEvent()

```
void FractalWidget::mouseMoveEvent (
    QMouseEvent * event ) [override], [protected]
```

Dessine le rectangle de sélection sur la fractale.

Paramètres

<i>event</i>	L'évènement de déplacement de la souris
--------------	---

6.4.2.4 mousePressEvent()

```
void FractalWidget::mousePressEvent (
    QMouseEvent * event ) [override], [protected]
```

Evènement de pression de la souris, permet de définir la zone de sélection de la fractale à afficher.

Paramètres

<i>event</i>	
--------------	--

6.4.2.5 mouseReleaseEvent()

```
void FractalWidget::mouseReleaseEvent (
    QMouseEvent * event ) [override], [protected]
```

Permet de définir la zone de sélection pour zoomer dans la fractale.

Paramètres

<i>event</i>	L'évènement de relâchement de la souris
--------------	---

6.4.2.6 newSelection

```
void Fractalium::FractalWidget::newSelection (
    const QPoint & ,
    const QPoint & ) [signal]
```

Voici le graphe des appelants de cette fonction :

6.4.2.7 setFractal()

```
void FractalWidget::setFractal (
    const QImage & image )
```

Permet de définir l'image de la fractale à afficher.

Paramètres

<i>image</i>	L'image de la fractale à afficher
--------------	-----------------------------------

La documentation de cette classe a été générée à partir des fichiers suivants :

- /Users/maximecolliat/Documents/Projet/Fractalium/src/[FractalWidget.hpp](#)
- /Users/maximecolliat/Documents/Projet/Fractalium/src/[FractalWidget.cpp](#)

6.5 Référence de la structure Fractalium::History

```
#include <History.hpp>
```

Graphe de collaboration de Fractalium::History:

Fonctions membres publiques

- `template<typename Archive >`
`void serialize (Archive &ar, const unsigned int version)`

Attributs publics

- `HistoryImage image`
- `std::pair< Fractalium::Double, Fractalium::Double > offset`
- `Fractalium::Double step_coord`

6.5.1 Documentation des fonctions membres

6.5.1.1 `serialize()`

```
template<typename Archive >
void Fractalium::History::serialize (
    Archive & ar,
    const unsigned int version ) [inline]
```

6.5.2 Documentation des données membres

6.5.2.1 `image`

`HistoryImage Fractalium::History::image`

6.5.2.2 `offset`

`std::pair<Fractalium::Double, Fractalium::Double> Fractalium::History::offset`

6.5.2.3 `step_coord`

`Fractalium::Double Fractalium::History::step_coord`

La documentation de cette structure a été générée à partir du fichier suivant :

- `/Users/maximecolliat/Documents/Projet/Fractalium/src/History.hpp`

6.6 Référence de la structure Fractalium::HistoryImage

```
#include <History.hpp>
```

Fonctions membres publiques

- `template<typename Archive > void serialize (Archive &ar, const unsigned int version)`
- `HistoryImage ()=default`
- `HistoryImage (const QImage &img)`
- `QImage toQImage () const`

Attributs publics

- `std::vector< std::vector< Color > > image {}`

6.6.1 Documentation des constructeurs et destructeur

6.6.1.1 HistoryImage() [1/2]

```
Fractalium::HistoryImage::HistoryImage ( ) [default]
```

6.6.1.2 HistoryImage() [2/2]

```
Fractalium::HistoryImage::HistoryImage (
    const QImage & img ) [inline]
```

6.6.2 Documentation des fonctions membres

6.6.2.1 serialize()

```
template<typename Archive >
void Fractalium::HistoryImage::serialize (
    Archive & ar,
    const unsigned int version ) [inline]
```

6.6.2.2 toQImage()

```
QImage Fractalium::HistoryImage::toQImage ( ) const [inline]
```

6.6.3 Documentation des données membres

6.6.3.1 image

```
std::vector<std::vector<Color> > Fractalium::HistoryImage::image {}
```

La documentation de cette structure a été générée à partir du fichier suivant :

— /Users/maximecolliat/Documents/Projet/Fractalium/src/[History.hpp](#)

6.7 Référence de la classe Fractalium::Image

```
#include <Image.hpp>
```

Fonctions membres publiques

— [Image](#) ()=default

— [Image](#) ([Image](#) &&other) noexcept

Constructeur par copie.

— [Image](#) & [operator=](#) (const [Image](#) &other)

Surcharge de l'opérateur = pour la copie.

— [Image](#) (uint32_t [width](#), uint32_t [height](#))

Constructeur de la classe [Image](#).

— void [setPixel](#) (uint16_t x, uint16_t y, int value)

Affectation d'une valeur à un pixel.

— void [merge](#) ([Image](#) &image)

Combine l'image courante avec une autre image.

— int [getPixel](#) (int x, int y)

— uint16_t [width](#) () const

— uint16_t [height](#) () const

Amis

- class [boost::serialization::access](#)
- `std::ostream & operator<< (std::ostream &os, const Image &image)`

6.7.1 Documentation des constructeurs et destructeur

6.7.1.1 [Image\(\)](#) [1/3]

```
Fractalium::Image::Image ( ) [default]
```

6.7.1.2 [Image\(\)](#) [2/3]

```
Fractalium::Image::Image (
    Fractalium::Image && other ) [noexcept]
```

Constructeur par copie.

Paramètres

<i>other</i>	
--------------	--

6.7.1.3 [Image\(\)](#) [3/3]

```
Fractalium::Image::Image (
    uint32_t size_x,
    uint32_t size_y ) [explicit]
```

Constructeur de la classe [Image](#).

Paramètres

<i>size_x</i>	taille de l'image
<i>size_y</i>	largeur de l'image

6.7.2 Documentation des fonctions membres

6.7.2.1 getPixel()

```
int Fractalium::Image::getPixel (
    int x,
    int y ) [inline]
```

6.7.2.2 height()

```
uint16_t Fractalium::Image::height ( ) const [inline]
```

Voici le graphe des appelants de cette fonction :

6.7.2.3 merge()

```
void Fractalium::Image::merge (
    Fractalium::Image & image )
```

Combine l'image courante avec une autre image.

Paramètres

<i>image</i>	
--------------	--

Voici le graphe d'appel pour cette fonction : Voici le graphe des appelants de cette fonction :

6.7.2.4 operator=()

```
Fractalium::Image & Fractalium::Image::operator= (
    const Image & other )
```

Surcharge de l'opérateur = pour la copie.

Paramètres

<i>other</i>	
--------------	--

Renvoie

[Image&](#)

6.7.2.5 setPixel()

```
void Fractalium::Image::setPixel (
    uint16_t x,
    uint16_t y,
    int value )
```

Affectation d'une valeur à un pixel.

Paramètres

<i>x</i>	
<i>y</i>	
<i>value</i>	

Voici le graphe des appelants de cette fonction :

6.7.2.6 width()

```
uint16_t Fractalium::Image::width ( ) const [inline]
```

Voici le graphe des appelants de cette fonction :

6.7.3 Documentation des fonctions amies et associées

6.7.3.1 boost::serialization::access

```
friend class boost::serialization::access [friend]
```

6.7.3.2 operator<<

```
std::ostream & operator<< (
    std::ostream & os,
    const Image & image ) [friend]
```

La documentation de cette classe a été générée à partir des fichiers suivants :

- /Users/maximecolliat/Documents/Projet/Fractalium/src/[Image.hpp](#)
- /Users/maximecolliat/Documents/Projet/Fractalium/src/[Image.cpp](#)

6.8 Référence de la classe MainWindow

```
#include <mainwindow.hpp>
```

Graphe d'héritage de MainWindow:

Graphe de collaboration de MainWindow:

Fonctions membres publiques

- `MainWindow` (`QWidget *parent=nullptr`)
- `~MainWindow` () `override=default`

Attributs publics statiques

- `static const int TOTAL_COLORS = 500`
- `static MainWindow * instance = nullptr`

Fonctions membres protégées

- `bool event` (`QEvent *event`) `override`

Gère les événements de la fenêtre.

6.8.1 Documentation des constructeurs et destructeur

6.8.1.1 MainWindow()

```
MainWindow::MainWindow (
    QWidget * parent = nullptr ) [explicit]
```

Voici le graphe d'appel pour cette fonction :

6.8.1.2 ~MainWindow()

```
MainWindow::~MainWindow ( ) [override], [default]
```

6.8.2 Documentation des fonctions membres

6.8.2.1 event()

```
bool MainWindow::event (
    QEvent * event ) [override], [protected]
```

Gère les événements de la fenêtre.

Paramètres

<i>event</i>	
--------------	--

Renvoie

Voici le graphe d'appel pour cette fonction : Voici le graphe des appelants de cette fonction :

6.8.3 Documentation des données membres

6.8.3.1 instance

```
MainWindow * MainWindow::instance = nullptr [static]
```

6.8.3.2 TOTAL_COLORS

```
const int MainWindow::TOTAL_COLORS = 500 [static]
```

La documentation de cette classe a été générée à partir des fichiers suivants :

- /Users/maximecolliat/Documents/Projet/Fractalium/src/[mainwindow.hpp](#)
- /Users/maximecolliat/Documents/Projet/Fractalium/src/[mainwindow.cpp](#)

6.9 Référence de la classe Fractalium::MPICalculator

```
#include <MPI.hpp>
```

Graphe de collaboration de Fractalium::MPICalculator:

Fonctions membres publiques

- [MPICalculator](#) (uint16_t [rank](#))

Constructeur de la classe [MPICalculator](#).

Fonctions membres publiques statiques

- static void `calculate` (const `MPIStruct` &`mpi_struct`, `Image` &`image`)

Calcul de l'image fractale en fonction des données reçues par le processus MPI et envoi de l'image calculée au processus 0 (master)

- static void `send` (const `MPIStruct` &`mpi_struct`, `Image` &`image`)

master envoie des taches à traiter aux noeuds enfants

- static void `receive` (`Image` &`image`)

Réception des données envoyées par le processus 0 (master) et calcul de l'image fractale.

- static void `stop` ()

Ferme le programme et arrete tous les noeuds.

Attributs publics statiques

- static boost::signals2::signal< void()> `finshed`
- static boost::signals2::signal< void(uint32_t)> `node_recived`
- static `MPIStruct` `mpi_struct`
- static uint16_t `rank` = 0
- static bool `is_running` = false
- static uint32_t `node_count` = 0
- static uint32_t `node_working` = 0

6.9.1 Documentation des constructeurs et destructeur

6.9.1.1 MPICalculator()

```
Fractalium::MPICalculator::MPICalculator (
    uint16_t rank ) [explicit]
```

Constructeur de la classe `MPICalculator`.

Paramètres

<code>rank</code>	Rang du processus MPI
-------------------	-----------------------

6.9.2 Documentation des fonctions membres

6.9.2.1 calculate()

```
void Fractalium::MPICalculator::calculate (
    const MPIStruct & data,
    Image & image ) [static]
```

Calcul de l'image fractale en fonction des données reçues par le processus MPI et envoi de l'image calculée au processus 0 (master)

Paramètres

<i>data</i>	Données reçues par le processus MPI
<i>image</i>	Image fractale

Voici le graphe d'appel pour cette fonction :

6.9.2.2 receive()

```
void Fractalium::MPICalculator::receive (
    Image & image ) [static]
```

Réception des données envoyées par le processus 0 (master) et calcul de l'image fractale.

Paramètres

<i>image</i>	
--------------	--

Voici le graphe d'appel pour cette fonction :

6.9.2.3 send()

```
void Fractalium::MPICalculator::send (
    const MPIStruct & data,
    Image & image ) [static]
```

master envoie des taches à traiter aux noeuds enfants

Paramètres

<i>data</i>	Données à envoyer
<i>image</i>	Image fractale

6.9.2.4 stop()

```
void Fractalium::MPICalculator::stop ( ) [static]
```

Ferme le programme et arrete tous les noeuds.

Voici le graphe des appelants de cette fonction :

6.9.3 Documentation des données membres

6.9.3.1 finshed

```
boost::signals2::signal< void()> Fractalium::MPICalculator::finshed [static]
```

6.9.3.2 is_running

```
bool Fractalium::MPICalculator::is_running = false [static]
```

6.9.3.3 mpi_struct

```
Fractalium::MPIStruct Fractalium::MPICalculator::mpi_struct [static]
```

6.9.3.4 node_count

```
uint32_t Fractalium::MPICalculator::node_count = 0 [static]
```

6.9.3.5 node_recived

```
boost::signals2::signal< void(uint32_t)> Fractalium::MPICalculator::node_recived [static]
```

6.9.3.6 node_working

```
uint32_t Fractalium::MPICalculator::node_working = 0 [static]
```

6.9.3.7 rank

```
uint16_t Fractalium::MPICalculator::rank = 0 [static]
```

La documentation de cette classe a été générée à partir des fichiers suivants :

- /Users/maximecolliat/Documents/Projet/Fractalium/src/[MPI.hpp](#)
- /Users/maximecolliat/Documents/Projet/Fractalium/src/[MPI.cpp](#)

6.10 Référence de la classe Fractalium::MPIStruct

```
#include <MPI.hpp>
```

Graphe de collaboration de Fractalium::MPIStruct:

Fonctions membres publiques

- [MPIStruct](#) (const [MPIStruct](#) &aStruct)
- [MPIStruct](#) (uint16_t [start_x](#), uint16_t [end_x](#), uint16_t [start_y](#), uint16_t [end_y](#), [Double](#) [offset_x](#), [Double](#) [offset_y](#), uint16_t [width](#), uint16_t [height](#), int [iterations](#), [Double](#) [step_coord](#), [Fractal](#) [fractal](#))
- [MPIStruct](#) ()=default

Attributs publics

- uint16_t [start_x](#)
- uint16_t [end_x](#)
- uint16_t [start_y](#)
- uint16_t [end_y](#)
- [Double](#) [offset_x](#)
- [Double](#) [offset_y](#)
- uint16_t [width](#)
- uint16_t [height](#)
- int [iterations](#) {}
- [Double](#) [step_coord](#)
- [Fractal](#) [fractal](#)

Amis

- class [boost::serialization::access](#)

6.10.1 Documentation des constructeurs et destructeur

6.10.1.1 MPIStruct() [1/3]

```
Fractalium::MPIStruct::MPIStruct (
    const MPIStruct & aStruct ) [inline]
```

6.10.1.2 MPIStruct() [2/3]

```
Fractalium::MPIStruct::MPIStruct (
    uint16_t start_x,
    uint16_t end_x,
    uint16_t start_y,
    uint16_t end_y,
    Double offset_x,
    Double offset_y,
    uint16_t width,
    uint16_t height,
    int iterations,
    Double step_coord,
    Fractal fractal ) [inline]
```

6.10.1.3 MPIStruct() [3/3]

```
Fractalium::MPIStruct::MPIStruct ( ) [default]
```

6.10.2 Documentation des fonctions amies et associées

6.10.2.1 boost::serialization::access

```
friend class boost::serialization::access [friend]
```

6.10.3 Documentation des données membres

6.10.3.1 end_x

```
uint16_t Fractalium::MPIStruct::end_x
```

6.10.3.2 end_y

```
uint16_t Fractalium::MPIStruct::end_y
```

6.10.3.3 fractal

```
Fractal Fractalium::MPIStruct::fractal
```

6.10.3.4 height

```
uint16_t Fractalium::MPIStruct::height
```

6.10.3.5 iterations

```
int Fractalium::MPIStruct::iterations {}
```

6.10.3.6 offset_x

```
Double Fractalium::MPIStruct::offset_x
```

6.10.3.7 offset_y

```
Double Fractalium::MPIStruct::offset_y
```

6.10.3.8 start_x

```
uint16_t Fractalium::MPIStruct::start_x
```

6.10.3.9 start_y

```
uint16_t Fractalium::MPIStruct::start_y
```

6.10.3.10 step_coord

```
Double Fractalium::MPIStruct::step_coord
```

6.10.3.11 width

```
uint16_t Fractalium::MPIStruct::width
```

La documentation de cette classe a été générée à partir du fichier suivant :

— /Users/maximecolliat/Documents/Projet/Fractalium/src/[MPI.hpp](#)

6.11 Référence de la classe PaintFractalEvent

```
#include <mainwindow.hpp>
```

Graphe d'héritage de PaintFractalEvent:

Graphe de collaboration de PaintFractalEvent:

Fonctions membres publiques

- [PaintFractalEvent](#) ()
- [~PaintFractalEvent](#) () override=default

Attributs publics statiques

- static const QEvent::Type [PaintFractalEventType](#) = static_cast<QEvent::Type>(QEvent::User + 1)

6.11.1 Documentation des constructeurs et destructeur**6.11.1.1 PaintFractalEvent()**

```
PaintFractalEvent::PaintFractalEvent ( ) [inline], [explicit]
```

6.11.1.2 ~PaintFractalEvent()

```
PaintFractalEvent::~~PaintFractalEvent ( ) [override], [default]
```

6.11.2 Documentation des données membres

6.11.2.1 PaintFractalEventType

```
const QEvent::Type PaintFractalEvent::PaintFractalEventType = static_cast<QEvent::Type> (QEvent←  
::User + 1) [static]
```

La documentation de cette classe a été générée à partir du fichier suivant :

— /Users/maximecolliat/Documents/Projet/Fractalium/src/[mainwindow.hpp](#)

6.12 Référence de la structure Fractalium::Settings

```
#include <Settings.hpp>
```

Types publics

— enum [CalculationType](#) : uint8_t { [COLLUMNS](#) , [SQUARES](#) }

Fonctions membres publiques statiques

— static void [saveSettings](#) ()

— static void [init](#) ()

Fonction qui initialise les paramètres de l'application à partir des paramètres sauvegardés dans le fichier de configuration de l'application.

— static void [resetCrash](#) ()

Fonction qui permet de réinitialiser les paramètres de l'application.

— static void [setCrash](#) ()

Fonction qui permet de définir que l'application a crashé

Attributs publics statiques

- static uint16_t `DISPLAY_SIZE_WIDTH` = 500
- static uint16_t `DISPLAY_SIZE_HEIGHT` = 500
- static uint32_t `ITERATIONS` = 100
- static uint32_t `NODES` = 0
- static bool `AUTO_SNAPSHOTS` = false
- static std::string `SAVE_PATH` = "./"
- static bool `IS_CRASHED` = false
- static const std::string `CRASH_SNAP_PATH` = "./snapshot_crash.fractalium"
- static `CalculationType` `CALCULATION_TYPE` = `Settings::CalculationType::COLLUMNS`
- static const std::string `GITHUB_URL` = "https://github.com/Sudo-Rahman/Fractalium"

6.12.1 Documentation des énumérations membres

6.12.1.1 CalculationType

```
enum Fractalium::Settings::CalculationType : uint8_t
```

Valeurs énumérées

COLLUMNS	
SQUARES	

6.12.2 Documentation des fonctions membres

6.12.2.1 init()

```
void Settings::init ( ) [static]
```

Fonction qui initialise les paramètres de l'application à partir des paramètres sauvegardés dans le fichier de configuration de l'application.

6.12.2.2 resetCrash()

```
void Settings::resetCrash ( ) [static]
```

Fonction qui permet de réinitialiser les paramètres de l'application.

6.12.2.3 saveSettings()

```
void Settings::saveSettings ( ) [static]
```

6.12.2.4 setCrash()

```
void Settings::setCrash ( ) [static]
```

Fonction qui permet de définir que l'application a crashé

6.12.3 Documentation des données membres

6.12.3.1 AUTO_SNAPSHOTS

```
bool Settings::AUTO_SNAPSHOTS = false [static]
```

6.12.3.2 CALCULATION_TYPE

```
Settings::CalculationType Settings::CALCULATION_TYPE = Settings::CalculationType::COLUMNS  
[static]
```

6.12.3.3 CRASH_SNAP_PATH

```
const std::string Settings::CRASH_SNAP_PATH = "./snapshot_crash.fractalium" [static]
```

6.12.3.4 DISPLAY_SIZE_HEIGHT

```
uint16_t Settings::DISPLAY_SIZE_HEIGHT = 500 [static]
```

6.12.3.5 DISPLAY_SIZE_WIDTH

```
uint16_t Settings::DISPLAY_SIZE_WIDTH = 500 [static]
```

6.12.3.6 GITHUB_URL

```
const std::string Settings::GITHUB_URL = "https://github.com/Sudo-Rahman/Fractalium" [static]
```

6.12.3.7 IS_CRASHED

```
bool Settings::IS_CRASHED = false [static]
```

6.12.3.8 ITERATIONS

```
uint32_t Settings::ITERATIONS = 100 [static]
```

6.12.3.9 NODES

```
uint32_t Settings::NODES = 0 [static]
```

6.12.3.10 SAVE_PATH

```
std::string Settings::SAVE_PATH = "." [static]
```

La documentation de cette structure a été générée à partir des fichiers suivants :

- /Users/maximecolliat/Documents/Projet/Fractalium/src/[Settings.hpp](#)
 - /Users/maximecolliat/Documents/Projet/Fractalium/src/[Settings.cpp](#)
-

6.13 Référence de la classe SettingsDialog

```
#include <SettingsDialog.hpp>
```

Graphe d'héritage de SettingsDialog:

Graphe de collaboration de SettingsDialog:

Fonctions membres publiques

— [SettingsDialog](#) (QWidget *parent=nullptr)

Constructeur de la classe [SettingsDialog](#).

6.13.1 Documentation des constructeurs et destructeur

6.13.1.1 SettingsDialog()

```
SettingsDialog::SettingsDialog (
    QWidget * parent = nullptr ) [explicit]
```

Constructeur de la classe [SettingsDialog](#).

Paramètres

<i>parent</i>	Widget parent
---------------	---------------

La documentation de cette classe a été générée à partir des fichiers suivants :

- /Users/maximecolliat/Documents/Projet/Fractalium/src/[SettingsDialog.hpp](#)
- /Users/maximecolliat/Documents/Projet/Fractalium/src/[SettingsDialog.cpp](#)

6.14 Référence de la classe SnapshotDialog

```
#include <SnapshotDialog.hpp>
```

Graphe d'héritage de SnapshotDialog:

Graphe de collaboration de SnapshotDialog:

Types publics

— enum class [Return](#) { [Import](#) , [Make](#) , [None](#) }

Fonctions membres publiques

— [SnapshotDialog](#) ([Fractalium::SnapshotHistory](#) &snapshotHistory, QWidget *parent)

Constructeur de [SnapshotDialog](#) qui permet de faire un instantané ou d'en importer un.

— const [Return](#) & [returnType](#) () const

6.14.1 Documentation des énumérations membres

6.14.1.1 Return

```
enum class SnapshotDialog::Return [strong]
```

Valeurs énumérées

Import	
Make	
None	

6.14.2 Documentation des constructeurs et destructeur

6.14.2.1 SnapshotDialog()

```
SnapshotDialog::SnapshotDialog (  
    Fractalium::SnapshotHistory & snapshotHistory,  
    QWidget * parent ) [explicit]
```

Constructeur de [SnapshotDialog](#) qui permet de faire un instantané ou d'en importer un.

Paramètres

<i>snapshotHistory</i>	
<i>parent</i>	

6.14.3 Documentation des fonctions membres

6.14.3.1 returnType()

```
const Return & SnapshotDialog::returnType ( ) const [inline]
```

La documentation de cette classe a été générée à partir des fichiers suivants :

- [/Users/maximecolliat/Documents/Projet/Fractalium/src/SnapshotDialog.hpp](#)
- [/Users/maximecolliat/Documents/Projet/Fractalium/src/SnapshotDialog.cpp](#)

6.15 Référence de la structure Fractalium::SnapshotHistory

```
#include <History.hpp>
```

Graphe de collaboration de Fractalium::SnapshotHistory:

Fonctions membres publiques

- [SnapshotHistory](#) ()=default
- [SnapshotHistory](#) (const std::vector< [History](#) > &history_back, const std::vector< [History](#) > &history_front, const [Fractal](#) &fractal)
- template<typename Archive >
void [serialize](#) (Archive &ar, const unsigned int version)

Attributs publics

- std::vector< [History](#) > history_back
- std::vector< [History](#) > history_front
- [Fractal](#) fractal

6.15.1 Documentation des constructeurs et destructeur

6.15.1.1 SnapshotHistory() [1/2]

```
Fractalium::SnapshotHistory::SnapshotHistory ( ) [default]
```

6.15.1.2 SnapshotHistory() [2/2]

```
Fractalium::SnapshotHistory::SnapshotHistory (
    const std::vector< History > & history_back,
    const std::vector< History > & history_front,
    const Fractal & fractal ) [inline]
```

6.15.2 Documentation des fonctions membres

6.15.2.1 serialize()

```
template<typename Archive >
void Fractalium::SnapshotHistory::serialize (
    Archive & ar,
    const unsigned int version ) [inline]
```

6.15.3 Documentation des données membres

6.15.3.1 fractal

```
Fractal Fractalium::SnapshotHistory::fractal
```

6.15.3.2 history_back

```
std::vector<History> Fractalium::SnapshotHistory::history_back
```

6.15.3.3 history_front

```
std::vector<History> Fractalium::SnapshotHistory::history_front
```

La documentation de cette structure a été générée à partir du fichier suivant :

— /Users/maximecolliat/Documents/Projet/Fractalium/src/[History.hpp](#)

Chapitre 7

Documentation des fichiers

7.1 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Color.hpp

`#include <boost/serialization/serialization.hpp>`

Graphe des dépendances par inclusion de Color.hpp:

7.2 Color.hpp

[Aller à la documentation de ce fichier.](#)

```
00001 //
00002 // Created by sr-71 on 19/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_COLOR_HPP
00006 #define FRACTALIUM_COLOR_HPP
00007
00008 #include <boost/serialization/serialization.hpp>
00009
00010
00011 namespace Fractalium
00012 {
00013     struct Color {
00014         int r, g, b;
00015     };
00016
00017     template<typename Archive>
00018     void serialize(Archive &ar, Color &color, const unsigned int version)
00019     {
00020         ar & color.r;
00021         ar & color.g;
00022         ar & color.b;
00023     }
00024 }
00025
00026 #endif //FRACTALIUM_COLOR_HPP
```

7.3 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Complex.cpp

`#include "Complex.hpp"`

Graphe des dépendances par inclusion de Complex.cpp:

7.4 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Complex.hpp

```
#include <Double.hpp>
```

Graphe des dépendances par inclusion de Complex.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [Fractalium::Complex](#)

Espaces de nommage

— namespace [Fractalium](#)

7.5 Complex.hpp

[Aller à la documentation de ce fichier.](#)

```
00001
00002 #ifndef FRACTALIUM_COMPLEX_HPP
00003 #define FRACTALIUM_COMPLEX_HPP
00004
00005
00006 #include <Double.hpp>
00007
00008 namespace Fractalium
00009 {
00010
00011     class Complex {
00012     public:
00013         Double real, imag;
00014
00015         Complex(Double r, Double i);
00016
00017         Complex();
00018
00019         Complex &operator=(Complex const &obj);
00020
00021         Complex operator+(Complex const &obj) const;
00022
00023         Complex operator-(Complex const &obj) const;
00024
00025         Complex operator*(Complex const &obj) const;
00026
00027         Complex operator/(Complex const &obj) const;
00028
00029         [[nodiscard]] Double norm() const;
00030     };
00031 }
00032 #endif //FRACTALIUM_COMPLEX_HPP
```

7.6 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Double.hpp

```
#include <boost/multiprecision/cpp_dec_float.hpp>
```

Graphe des dépendances par inclusion de Double.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Espaces de nommage

— namespace [Fractalium](#)

Définitions de type

— typedef boost::multiprecision::cpp_dec_float_100 [Fractalium::Double](#)

7.7 Double.hpp

[Aller à la documentation de ce fichier.](#)

```
00001 //
00002 // Created by sr-71 on 03/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_DOUBLE_HPP
00006 #define FRACTALIUM_DOUBLE_HPP
00007
00008 #include <boost/multiprecision/cpp_dec_float.hpp>
00009
00010 namespace Fractalium
00011 {
00012     typedef boost::multiprecision::cpp_dec_float_100 Double;
00013 }
00014
00015 #endif //FRACTALIUM_DOUBLE_HPP
```

7.8 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Fractal.cpp

```
#include "Fractal.hpp"
```

Graphe des dépendances par inclusion de Fractal.cpp:

7.9 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Fractal.hpp

```
#include <Complex.hpp>
```

```
#include <boost/serialization/serialization.hpp>
```

Graphe des dépendances par inclusion de Fractal.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [Fractalium::Fractal](#)

Espaces de nommage

— namespace [Fractalium](#)

7.10 Fractal.hpp

[Aller à la documentation de ce fichier.](#)

```

00001 //
00002 // Created by sr-71 on 28/11/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_FRACTAL_HPP
00006 #define FRACTALIUM_FRACTAL_HPP
00007
00008 #include <Complex.hpp>
00009 #include <boost/serialization/serialization.hpp>
00010
00011
00012 namespace Fractalium
00013 {
00014
00015
00016     class Fractal {
00017
00018
00019     public:
00020         enum class FractalType {
00021             Mandelbrot,
00022             Julia,
00023             BurningShip,
00024             Newton,
00025             Koch
00026         };
00027
00028         explicit Fractal(FractalType type = FractalType::Mandelbrot) : _type(type)
00029         {}
00030
00031         [[nodiscard]] int pointCheck(const Complex &pointm, int iterations) const;
00032
00033         void setType(FractalType type)
00034         {
00035             _type = type;
00036         }
00037
00038         [[nodiscard]] FractalType getType() const
00039         {
00040             return _type;
00041         }
00042
00043     private:
00044
00045         FractalType _type;
00046
00047
00048         friend class boost::serialization::access;
00049
00050         template<class Archive>
00051         void serialize(Archive &ar, const unsigned int version)
00052         {
00053             ar & _type;
00054         }
00055
00056         class Mandelbrot {
00057     public:
00058             [[nodiscard]] static int pointCheck(const Complex &point, int iterations);
00059         };
00060
00061         class Julia {
00062     public:
00063             static Complex juliaConstant;
00064             [[nodiscard]] static int pointCheck(const Complex &point, int iterations);
00065         };
00066
00067         class BurningShip {
00068     public:
00069             [[nodiscard]] static int pointCheck(const Complex &point, int iterations);
00070         };
00071
00072         class Newton {
00073     public:
00074             [[nodiscard]] static int pointCheck(const Complex &point, int iterations);
00075         };
00076
00077         class Koch {
00078     public:
00079             [[nodiscard]] static int pointCheck(const Complex &point, int iterations);
00080         };
00081
00082     };

```

```
00083 }
00084
00085 #endif //FRACTALIUM_FRACTAL_HPP
```

7.11 Référence du fichier /Users/maximecolliat/Documents/Projet/↵ Fractalium/src/FractalWidget.cpp

```
#include "FractalWidget.hpp"
#include <QMouseEvent>
```

Graphe des dépendances par inclusion de FractalWidget.cpp:

7.12 Référence du fichier /Users/maximecolliat/Documents/Projet/↵ Fractalium/src/FractalWidget.hpp

```
#include <QLabel>
#include <QRubberBand>
```

Graphe des dépendances par inclusion de FractalWidget.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [Fractalium::FractalWidget](#)

Espaces de nommage

— namespace [Fractalium](#)

7.13 FractalWidget.hpp

[Aller à la documentation de ce fichier.](#)

```
00001 //
00002 // Created by sr-71 on 02/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_FRACTALWIDGET_HPP
00006 #define FRACTALIUM_FRACTALWIDGET_HPP
00007
00008
00009 #include <QLabel>
00010 #include <QRubberBand>
00011
00012 namespace Fractalium
00013 {
00014
00015     class FractalWidget : public QLabel {
00016     Q_OBJECT
00017
00018         bool _rubber_band_selection = false;
00019
00020         QRubberBand *_rubber_band;
00021
00022         QPoint _start_point, _release_point;
00023
00024         bool _selection_enabled{true};
```

```

00025
00026     void setPixmap(const QPixmap &pixmap);
00027
00028     public:
00029         explicit FractalWidget(QWidget *parent = nullptr);
00030
00031         ~FractalWidget() override
00032         = default;
00033
00034         void setFractal(const QImage &image);
00035
00036         void disableSelection()
00037         { _selection_enabled = false; }
00038
00039         void enableSelection()
00040         { _selection_enabled = true; }
00041
00042     signals:
00043
00044         void newSelection(const QPoint &, const QPoint &);
00045
00046     protected:
00047         void mousePressEvent(QMouseEvent *event) override;
00048
00049         void mouseReleaseEvent(QMouseEvent *event) override;
00050
00051         void mouseMoveEvent(QMouseEvent *event) override;
00052
00053     };
00054 }
00055
00056 #endif //FRACTALIUM_FRACTALWIDGET_HPP

```

7.14 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/History.hpp

```

#include <QImage>
#include <boost/serialization/serialization.hpp>
#include <boost/serialization/vector.hpp>
#include <boost/serialization/utility.hpp>
#include <fstream>
#include <boost/archive/text_oarchive.hpp>
#include <boost/archive/text_iarchive.hpp>
#include <Double.hpp>
#include <Fractal.hpp>
#include <Color.hpp>

```

Graphe des dépendances par inclusion de History.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- struct [Fractalium::HistoryImage](#)
- struct [Fractalium::History](#)
- struct [Fractalium::SnapshotHistory](#)

Espaces de nommage

- namespace [Fractalium](#)

7.15 History.hpp

[Aller à la documentation de ce fichier.](#)

```

00001 //
00002 // Created by sr-71 on 19/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_HISTORY_HPP
00006 #define FRACTALIUM_HISTORY_HPP
00007
00008 #include <QImage>
00009 #include <boost/serialization/serialization.hpp>
00010 #include <boost/serialization/vector.hpp>
00011 #include <boost/serialization/utility.hpp>
00012 #include <fstream>
00013 #include <boost/archive/text_oarchive.hpp>
00014 #include <boost/archive/text_iarchive.hpp>
00015 #include <Double.hpp>
00016 #include <Fractal.hpp>
00017 #include <Color.hpp>
00018
00019 namespace Fractalium
00020 {
00021     struct HistoryImage {
00022         std::vector<std::vector<Color>> image{};
00023
00024         template<typename Archive>
00025         void serialize(Archive &ar, const unsigned int version)
00026         {
00027             ar & image;
00028         }
00029
00030         HistoryImage() = default;
00031
00032         HistoryImage(const QImage &img)
00033         {
00034             image = std::vector<std::vector<Color>>(img.height(), std::vector<Color>(img.width()));
00035
00036             for (int i = 0; i < img.height(); ++i)
00037             {
00038                 for (int j = 0; j < img.width(); ++j)
00039                 {
00040                     QColor color = img.pixelColor(j, i);
00041                     image[i][j].r = color.red();
00042                     image[i][j].g = color.green();
00043                     image[i][j].b = color.blue();
00044                 }
00045             }
00046         }
00047
00048         [[nodiscard]] QImage toQImage() const
00049         {
00050             QImage img(image[0].size(), image.size(), QImage::Format_RGB32);
00051
00052             for (int i = 0; i < image.size(); ++i)
00053             {
00054                 for (int j = 0; j < image[0].size(); ++j)
00055                 {
00056                     img.setPixelColor(j, i, QColor(image[i][j].r, image[i][j].g, image[i][j].b));
00057                 }
00058             }
00059
00060             return img;
00061         }
00062     };
00063
00064
00065     struct History {
00066         HistoryImage image;
00067         std::pair<Fractalium::Double, Fractalium::Double> offset;
00068         Fractalium::Double step_coord;
00069
00070         template<typename Archive>
00071         void serialize(Archive &ar, const unsigned int version)
00072         {
00073             ar & image;
00074             ar & offset;
00075             ar & step_coord;
00076         }
00077     };
00078
00079     struct SnapshotHistory {
00080         std::vector<History> history_back;
00081         std::vector<History> history_front;
00082         Fractal fractal;

```

```

00083
00084     SnapshotHistory() = default;
00085
00086     SnapshotHistory(const std::vector<History> &history_back, const std::vector<History>
&history_front,
00087                     const Fractal &fractal)
00088         : history_back(history_back), history_front(history_front), fractal(fractal)
00089     {}
00090
00091     template<typename Archive>
00092     void serialize(Archive &ar, const unsigned int version)
00093     {
00094         ar & history_back;
00095         ar & history_front;
00096         ar & fractal;
00097     }
00098 };
00099
00100
00106     static void makeSnapshot(const std::string &filename, const SnapshotHistory &snapshotHistory)
00107     {
00108         std::ofstream ofs(filename);
00109         boost::archive::text_oarchive oa(ofs);
00110         oa << snapshotHistory;
00111     }
00112
00118     static void importSnapshot(const std::string &filename, SnapshotHistory &snapshotHistory)
00119     {
00120         std::ifstream ifs(filename);
00121         boost::archive::text_iarchive ia(ifs);
00122         ia >> snapshotHistory;
00123     }
00124 }
00125
00126 #endif //FRACTALIUM_HISTORY_HPP

```

7.16 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Image.cpp

```

#include "Image.hpp"
#include <iostream>
#include <boost/mpi.hpp>

```

Graphe des dépendances par inclusion de Image.cpp:

7.17 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Image.hpp

```

#include <boost/serialization/serialization.hpp>
#include <boost/serialization/utility.hpp>
#include <boost/serialization/vector.hpp>

```

Graphe des dépendances par inclusion de Image.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [Fractalium::Image](#)

Espaces de nommage

— namespace [Fractalium](#)

7.18 Image.hpp

[Aller à la documentation de ce fichier.](#)

```

00001 //
00002 // Created by sr-71 on 04/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_IMAGE_HPP
00006 #define FRACTALIUM_IMAGE_HPP
00007
00008 #include <boost/serialization/serialization.hpp>
00009 #include <boost/serialization/utility.hpp>
00010 #include <boost/serialization/vector.hpp>
00011
00012 namespace Fractalium
00013 {
00014     class Image {
00015     public:
00016         std::vector<std::vector<int>> _image;
00017         uint32_t _width{};
00018         uint32_t _height{};
00019
00020         Image() = default;
00021         Image(Image &other) noexcept;
00022         Image &operator=(const Image &other);
00023         explicit Image(uint32_t width, uint32_t height);
00024
00025         void setPixel(uint16_t x, uint16_t y, int value);
00026         void merge(Image &image);
00027
00028         int getPixel(int x, int y)
00029         {
00030             return _image[x][y];
00031         }
00032
00033         [[nodiscard]] uint16_t width() const
00034         { return _width; }
00035
00036         [[nodiscard]] uint16_t height() const
00037         { return _height; }
00038
00039         friend std::ostream &operator<<(std::ostream &os, const Image &image);
00040
00041     private:
00042         friend class boost::serialization::access;
00043
00044         template<class Archive>
00045         void serialize(Archive &ar, const unsigned int version)
00046         {
00047             ar & _image;
00048             ar & _width;
00049             ar & _height;
00050         }
00051     };
00052
00053 #endif //FRACTALIUM_IMAGE_HPP

```

7.19 Référence du fichier /Users/maximecolliat/Documents/Projet/↵ Fractalium/src/mainwindow.cpp

```

#include "mainwindow.hpp"
#include <iostream>
#include <QApplication>
#include <MPI.hpp>
#include <QFileDialog>
#include <QLayout>
#include <QMessageBox>

```

```
#include <Color.hpp>
#include <SnapshotDialog.hpp>
#include <csignal>
#include <History.hpp>
#include <Settings.hpp>
#include <SettingsDialog.hpp>
#include <QDateTime>
```

Graphe des dépendances par inclusion de mainwindow.cpp:

Fonctions

- [Color getColorForDivergence](#) (int divergence, int maxDivergence, int red, int green, int blue)

Retourne la couleur en fonction de la divergence et de la couleur max du fractal (couleur dynamique)

Variables

- unsigned short [color_mode](#) = 0
- auto [get_color](#)
- [Color c](#) {}

7.19.1 Documentation des fonctions

7.19.1.1 getColorForDivergence()

```
Color getColorForDivergence (
    int divergence,
    int maxDivergence,
    int red,
    int green,
    int blue )
```

Retourne la couleur en fonction de la divergence et de la couleur max du fractal (couleur dynamique)

Paramètres

<i>divergence</i>	
<i>maxDivergence</i>	
<i>red</i>	
<i>green</i>	
<i>blue</i>	

Renvoie

7.19.2 Documentation des variables

7.19.2.1 c

```
Color c {}
```

7.19.2.2 color_mode

```
unsigned short color_mode = 0
```

7.19.2.3 get_color

```
auto get_color
```

7.20 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/mainwindow.hpp ↩

```
#include <QMainWindow>
#include <QColor>
#include <QTimer>
#include <QMouseEvent>
#include <QProgressBar>
#include <QImage>
#include <QMenuBar>
#include <QStatusBar>
#include <FractalWidget.hpp>
#include <Fractal.hpp>
#include <Image.hpp>
#include <History.hpp>
```

Graphe des dépendances par inclusion de mainwindow.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [MainWindow](#)

— class [PaintFractalEvent](#)

7.21 mainwindow.hpp

[Aller à la documentation de ce fichier.](#)

```

00001 #ifndef MANDELBROT_MAINWINDOW_H
00002 #define MANDELBROT_MAINWINDOW_H
00003
00004 #include <QMainWindow>
00005 #include <QColor>
00006 #include <QTimer>
00007 #include <QMouseEvent>
00008 #include <QProgressBar>
00009 #include <QImage>
00010 #include <QMenuBar>
00011 #include <QStatusBar>
00012 #include <FractalWidget.hpp>
00013 #include <Fractal.hpp>
00014 #include <Image.hpp>
00015 #include <History.hpp>
00016
00017
00018 class MainWindow : public QMainWindow {
00019     Q_OBJECT
00020
00021     QMenuBar *_menu_bar;
00022
00023     QStatusBar *_status_bar;
00024
00025     QImage *_image;
00026
00027     Fractalium::FractalWidget *_label;
00028
00029     std::vector<QColor> _color_map;
00030
00031     std::pair<Fractalium::Double, Fractalium::Double> _offset = {-2.1, -2};
00032
00033     Fractalium::Double _step_coord;
00034
00035     Fractalium::Fractal *_fractal;
00036
00037     Fractalium::Image _divergence_image;
00038
00039     std::vector<Fractalium::History> _back_history, _front_history;
00040
00041     QTimer *_timer;
00042     uint64_t _counter = 0;
00043     QLabel *_status_label;
00044
00045     QString _launched_time;
00046
00047 public:
00048     explicit MainWindow(QWidget *parent = nullptr);
00049
00050     ~MainWindow() override = default;
00051
00052     static const int TOTAL_COLORS = 500;
00053
00054     static MainWindow *instance;
00055
00056 private:
00057     void paintFractal();
00058
00059     void newSelection(const QPoint &start, const QPoint &end);
00060
00061     void setupUi();
00062
00063     void mpiCalculate();
00064
00065     void back();
00066
00067     void front();
00068
00069     void saveImage();
00070
00071     void updateColor();
00072
00073     void loadSnapshot(const Fractalium::SnapshotHistory &snapshot);
00074
00075     void lauchAfterCrash();
00076
00077     static void signalSnapshot(int signum, const std::vector<Fractalium::History> &backHistory,
00078                                const std::vector<Fractalium::History> &frontHistory, const
00079     Fractalium::Fractal &fractal);
00080
00081     static void handleSignal();

```



```

00082
00083 protected:
00084     bool event(QEvent *event) override;
00085 };
00086
00087 class PaintFractalEvent : public QEvent {
00088 public:
00089     static const QEvent::Type PaintFractalEventType = static_cast<QEvent::Type>(QEvent::User + 1);
00090
00091     explicit PaintFractalEvent() : QEvent(PaintFractalEventType)
00092     {}
00093
00094     ~PaintFractalEvent() override = default;
00095 };
00096
00097
00098 #endif //MANDELBROT_MAINWINDOW_H

```

7.22 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/MPI.cpp

```

#include "MPI.hpp"
#include <boost/mpi.hpp>
#include <Settings.hpp>

```

Graphe des dépendances par inclusion de MPI.cpp:

7.23 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/MPI.hpp

```

#include <iostream>
#include <vector>
#include <Double.hpp>
#include <Fractal.hpp>
#include <Image.hpp>
#include <future>
#include <boost/signals2.hpp>

```

Graphe des dépendances par inclusion de MPI.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

- class [Fractalium::MPIStruct](#)
- class [Fractalium::MPICalculator](#)

Espaces de nommage

- namespace [Fractalium](#)

7.24 MPI.hpp

[Aller à la documentation de ce fichier.](#)

```

00001 //
00002 // Created by sr-71 on 03/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_MPI_HPP
00006 #define FRACTALIUM_MPI_HPP
00007
00008 #include <iostream>
00009 #include <vector>
00010 #include <Double.hpp>
00011 #include <Fractal.hpp>
00012 #include <Image.hpp>
00013 #include <future>
00014 #include <boost/signals2.hpp>
00015
00016 namespace Fractalium
00017 {
00018     using Fractalium::Double;
00019
00020     class MPIStruct {
00021     public:
00022         MPIStruct(const MPIStruct &aStruct)
00023         {
00024             start_x = aStruct.start_x;
00025             end_x = aStruct.end_x;
00026             start_y = aStruct.start_y;
00027             end_y = aStruct.end_y;
00028             offset_x = aStruct.offset_x;
00029             offset_y = aStruct.offset_y;
00030             width = aStruct.width;
00031             height = aStruct.height;
00032             iterations = aStruct.iterations;
00033             step_coord = aStruct.step_coord;
00034             fractal = aStruct.fractal;
00035         }
00036
00037         uint16_t start_x;
00038         uint16_t end_x;
00039
00040         uint16_t start_y;
00041         uint16_t end_y;
00042
00043         Double offset_x;
00044         Double offset_y;
00045
00046         uint16_t width;
00047         uint16_t height;
00048
00049         int iterations{};
00050
00051         Double step_coord;
00052
00053         Fractal fractal;
00054
00055         MPIStruct(uint16_t start_x, uint16_t end_x, uint16_t start_y, uint16_t end_y, Double offset_x,
00056 Double offset_y,
00057         uint16_t width, uint16_t height, int iterations, Double step_coord, Fractal fractal)
00058 :
00059         start_x(start_x), end_x(end_x), start_y(start_y), end_y(end_y), offset_x(offset_x),
00060         offset_y(offset_y), width(width), height(height), iterations(iterations),
00061         step_coord(step_coord),
00062         fractal(fractal)
00063     {}
00064
00065     MPIStruct() = default;
00066
00067 private:
00068     friend class boost::serialization::access;
00069
00070     template<class Archive>
00071     void serialize(Archive &ar, const unsigned int version)
00072     {
00073         ar & start_x;
00074         ar & end_x;
00075         ar & start_y;
00076     }
00077
00078
00079

```

```

00080         ar & end_y;
00081         ar & offset_x;
00082         ar & offset_y;
00083         ar & width;
00084         ar & height;
00085         ar & iterations;
00086         ar & step_coord;
00087         ar & fractal;
00088     }
00089 };
00090
00091
00092 class MPICalculator {
00093
00094     static std::future<void> future;
00095
00096 public:
00097     static boost::signals2::signal<void()> finshed;
00098
00099     static boost::signals2::signal<void(uint32_t)> node_recived;
00100
00101     static MPIStruct mpi_struct;
00102
00103     static uint16_t rank;
00104
00105     static bool is_running;
00106
00107     static uint32_t node_count;
00108     static uint32_t node_working;
00109
00110     explicit MPICalculator(uint16_t rank);
00111
00112     static void calculate(const MPIStruct &mpi_struct, Image &image);
00113
00114     static void send(const MPIStruct &mpi_struct, Image &image);
00115
00116     static void receive(Image &image);
00117
00118
00119 public:
00120
00121     static void stop();
00122 };
00123
00124 }
00125
00126
00127 #endif //FRACTALIUM_MPI_HPP

```

7.25 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Settings.cpp

```

#include <Settings.hpp>
#include <QCoreApplication>
#include <QSettings>

```

Graphe des dépendances par inclusion de Settings.cpp:

7.26 Référence du fichier

/Users/maximecolliat/Documents/Projet/Fractalium/src/Settings.hpp

```

#include <cstdint>
#include <string>
#include <vector>

```

Graphe des dépendances par inclusion de Settings.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— struct [Fractalium::Settings](#)

Espaces de nommage

— namespace [Fractalium](#)

7.27 Settings.hpp

[Aller à la documentation de ce fichier.](#)

```
00001 //
00002 // Created by sr-71 on 30/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_SETTINGS_HPP
00006 #define FRACTALIUM_SETTINGS_HPP
00007
00008 #include <stdint>
00009 #include <string>
00010 #include <vector>
00011
00012 namespace Fractalium
00013 {
00014     struct Settings {
00015
00016         static uint16_t DISPLAY_SIZE_WIDTH;
00017         static uint16_t DISPLAY_SIZE_HEIGHT;
00018
00019         static uint32_t ITERATIONS;
00020
00021         static uint32_t NODES;
00022
00023         static bool AUTO_SNAPSHOTS;
00024         static std::string SAVE_PATH;
00025
00026         static bool IS_CRASHED;
00027         const static std::string CRASH_SNAP_PATH ;
00028
00029         enum CalculationType : uint8_t {
00030             COLUMNS,
00031             SQUARES,
00032         };
00033
00034         static CalculationType CALCULATION_TYPE;
00035
00036         static void saveSettings();
00037
00038         static void init();
00039
00040         static void resetCrash();
00041
00042         static void setCrash();
00043
00044         static const std::string GITHUB_URL;
00045     };
00046 }
00047
00048 #endif //FRACTALIUM_SETTINGS_HPP
```

7.28 Référence du fichier /Users/maximecolliat/Documents/Projet/↵ Fractalium/src/SettingsDialog.cpp

```
#include "SettingsDialog.hpp"
#include <QLayout>
#include <QFormLayout>
```

```
#include <QLabel>
#include <QCheckBox>
#include <QGroupBox>
#include <QPushButton>
#include <QLineEdit>
#include <QSpinBox>
#include <QSlider>
#include <QFileDialog>
#include <QComboBox>
```

Graphe des dépendances par inclusion de SettingsDialog.cpp:

7.29 Référence du fichier /Users/maximecolliat/Documents/Projet/Fractalium/src/SettingsDialog.hpp ↩

```
#include <QDialog>
#include <QVBoxLayout>
#include <Settings.hpp>
```

Graphe des dépendances par inclusion de SettingsDialog.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [SettingsDialog](#)

7.30 SettingsDialog.hpp

[Aller à la documentation de ce fichier.](#)

```
00001 //
00002 // Created by sr-71 on 30/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_SETTINGSDIALOG_HPP
00006 #define FRACTALIUM_SETTINGSDIALOG_HPP
00007
00008
00009 #include <QDialog>
00010 #include <QVBoxLayout>
00011 #include <Settings.hpp>
00012
00013
00014 class SettingsDialog : public QDialog {
00015     Q_OBJECT
00016
00017 public:
00018
00019     explicit SettingsDialog(QWidget *parent = nullptr);
00020
00021
00022 private:
00023     QSize _size;
00024     int _iterations;
00025     bool _auto_snapshots;
00026     QString _save_path;
00027     Fractalium::Settings::CalculationType _calculation_type;
00028
00029
00030     QVBoxLayout *_layout;
00031
00032     void initUi();
00033
00034     void initResizeUi();
00035 }
```

```

00036     void initIterationsUi();
00037
00038     void initAutoSnapshotsUi();
00039
00040     void initCalculationTypeUi();
00041
00042 };
00043
00044
00045 #endif //FRACTALIUM_SETTINGSDIALOG_HPP

```

7.31 Référence du fichier /Users/maximecolliat/Documents/Projet/↵ Fractalium/src/SnapshotDialog.cpp

```

#include "SnapshotDialog.hpp"
#include <QFileDialog>
#include <qgridlayout.h>
#include <QDateTime>

```

Graphe des dépendances par inclusion de SnapshotDialog.cpp:

7.32 Référence du fichier /Users/maximecolliat/Documents/Projet/↵ Fractalium/src/SnapshotDialog.hpp

```

#include <QPushButton>
#include <QDialog>
#include <History.hpp>

```

Graphe des dépendances par inclusion de SnapshotDialog.hpp: Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

— class [SnapshotDialog](#)

7.33 SnapshotDialog.hpp

[Aller à la documentation de ce fichier.](#)

```

00001 //
00002 // Created by sr-71 on 19/12/2023.
00003 //
00004
00005 #ifndef FRACTALIUM_SNAPSHOTDIALOG_HPP
00006 #define FRACTALIUM_SNAPSHOTDIALOG_HPP
00007
00008 #include <QPushButton>
00009 #include <QDialog>
00010 #include <History.hpp>
00011
00012 class SnapshotDialog : public QDialog {
00013     Q_OBJECT
00014
00015 public:
00016     explicit SnapshotDialog(Fractalium::SnapshotHistory &snapshotHistory, QWidget *parent);
00017
00018     enum class Return {
00019         Import,
00020         Make,
00021         None
00022     };

```

```
00023
00024     [[nodiscard]] const Return &returnType() const
00025     {
00026         return _return;
00027     }
00028
00029 private:
00030     QPushButton *_import_snapshot, *make_snapshot;
00031     Return _return{Return::None};
00032
00033 };
00034
00035
00036 #endif //FRACTALIUM_SNAPSHOTDIALOG_HPP
```
