

My Project

Generated by Doxygen 1.8.9.1

Wed Mar 16 2016 23:10:22

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	Application Class Reference	5
3.1.1	Member Function Documentation	7
3.1.1.1	DoForEns	7
3.1.1.2	DoForForme	7
3.1.1.3	DoForQPointF	7
3.1.1.4	getCentre	8
3.1.1.5	getK	8
3.1.1.6	getm11	8
3.1.1.7	getm12	8
3.1.1.8	getm21	8
3.1.1.9	getm22	8
3.1.1.10	getTheta	8
3.1.1.11	getv1	9
3.1.1.12	getv2	9
3.1.1.13	isHomothetie	9
3.1.1.14	isRotation	9
3.1.1.15	setCentre	9
3.1.1.16	setK	9
3.1.1.17	setm11	9
3.1.1.18	setm12	10
3.1.1.19	setm21	10
3.1.1.20	setm22	10
3.1.1.21	setv1	10
3.1.1.22	setv2	10
3.2	Forme Class Reference	10

3.2.1	Member Function Documentation	11
3.2.1.1	AddPoint	11
3.2.1.2	generateExisting	11
3.2.1.3	GetPoint	11
3.2.1.4	GetSize	11
3.3	Fractale Class Reference	12
3.3.1	Member Function Documentation	12
3.3.1.1	AddApplication	12
3.3.1.2	AddForme	12
3.3.1.3	AddHomothetie	13
3.3.1.4	AddHomothetie	13
3.3.1.5	AddHomothetie	13
3.3.1.6	AddRotation	13
3.3.1.7	AddRotation	13
3.3.1.8	AddRotation	13
3.3.1.9	generateExisting	14
3.3.1.10	getFromEnsForme	14
3.3.1.11	getSizeEnsAppli	14
3.3.1.12	getSizeEnsForme	14
3.3.1.13	isLikeCantor	14
3.3.1.14	setLikeCantor	15
3.4	Homothetie Class Reference	15
3.4.1	Constructor & Destructor Documentation	16
3.4.1.1	Homothetie	16
3.4.1.2	Homothetie	16
3.4.2	Member Function Documentation	16
3.4.2.1	setHomothetie	16
3.4.2.2	setHomothetie	16
3.5	Rotation Class Reference	17
3.5.1	Constructor & Destructor Documentation	17
3.5.1.1	Rotation	17
3.5.2	Member Function Documentation	18
3.5.2.1	setRotation	18
3.5.2.2	setRotation	18
3.6	SimilitudeDirecte Class Reference	18
3.6.1	Constructor & Destructor Documentation	19
3.6.1.1	SimilitudeDirecte	19
3.6.1.2	SimilitudeDirecte	19
3.6.2	Member Function Documentation	20
3.6.2.1	setSimilitudeDirecte	20

3.6.2.2	setSimilitudeDirecte	20
3.6.2.3	setTheta	20
3.7	SimilitudeIndirecte Class Reference	21
3.7.1	Constructor & Destructor Documentation	21
3.7.1.1	SimilitudeIndirecte	22
3.7.1.2	SimilitudeIndirecte	23
3.7.2	Member Function Documentation	23
3.7.2.1	setSimilitudeIndirecte	23
3.7.2.2	setSimilitudeIndirecte	23
3.8	Window Class Reference	24
3.8.1	Member Function Documentation	25
3.8.1.1	eventFilter	25
Index		27

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Application	5
Homothetie	15
Rotation	17
SimilitudeDirecte	18
SimilitudeIndirecte	21
Forme	10
Fractale	12
QMainWindow	
Window	24

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

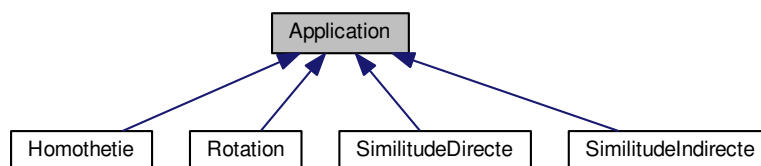
Application	5
Forme	10
Fractale	12
Homothetie	15
Rotation	17
SimilitudeDirecte	18
SimilitudeIndirecte	21
Window	24

Chapter 3

Class Documentation

3.1 Application Class Reference

Inheritance diagram for Application:



Public Member Functions

- [Application](#) ()
[Application::Application](#) Constructeur par default de [Application](#).
- void [setm11](#) (qreal m)
[Application::setm11](#).
- void [setm12](#) (qreal m)
[Application::setm12](#).
- void [setm21](#) (qreal m)
[Application::setm21](#).
- void [setm22](#) (qreal m)
[Application::setm22](#).
- void [setv1](#) (qreal m)
[Application::setv1](#).
- void [setv2](#) (qreal m)
[Application::setv2](#).
- void [setCentre](#) (QPointF P)
[Application::setCentre](#).
- void [setK](#) (qreal K)
[Application::setK](#).
- qreal [getm11](#) () const

- [Application::getm11.](#)
- qreal [getm12](#) () const
[Application::getm12.](#)
- qreal [getm21](#) () const
[Application::getm21.](#)
- qreal [getm22](#) () const
[Application::getm22.](#)
- qreal [getv1](#) () const
[Application::getv1.](#)
- qreal [getv2](#) () const
[Application::getv2.](#)
- QPointF [getCentre](#) () const
[Application::getCentre.](#)
- qreal [getK](#) () const
[Application::getK.](#)
- qreal [getTheta](#) () const
[Application::getTheta.](#)
- bool [isHomothetie](#) () const
[Application::isHomothetie.](#)
- bool [isRotation](#) () const
[Application::isRotation.](#)
- QPointF [DoForQPointF](#) (QPointF const &P) const
[Application::DoForQPointF.](#)
- [Forme DoForForme](#) ([Forme](#) const &F) const
[Application::DoForForme.](#)
- QList< [Forme](#) > [DoForEns](#) (const QList< [Forme](#) > &EnsForme) const
[Application::DoForEns.](#)

Protected Attributes

- qreal [k](#)
k Rapport de l'application
- qreal [m11](#)
Décrit la matrice de rotation

$$\begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \text{ ou } \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix}$$

- qreal [m12](#)
Décrit la matrice de rotation

$$\begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \text{ ou } \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix}$$

- qreal [m21](#)
Décrit la matrice de rotation

$$\begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \text{ ou } \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix}$$

- qreal [m22](#)

Décrit la matrice de rotation

$$\begin{pmatrix} m_{1,1} & m_{1,2} \\ m_{2,1} & m_{2,2} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \text{ ou } \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix}$$

- qreal [v1](#)

v1 Parametre du vecteur de translation V

$$\mathbf{V} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

- qreal [v2](#)

v2 Parametre du vecteur de translation V

$$\mathbf{V} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

- QPointF [Centre](#)

Centre de l'application.

3.1.1 Member Function Documentation

3.1.1.1 QList< **Forme** > Application::DoForEns (const QList< **Forme** > & *EnsForme*) const

[Application::DoForEns.](#)

Parameters

<i>EnsForme</i>	
-----------------	--

Returns

Image de l'application sur chaque des forme

3.1.1.2 **Forme** Application::DoForForme (**Forme** const & *F*) const

[Application::DoForForme.](#)

Parameters

<i>F</i>	Forme
----------	-----------------------

Returns

Image de l'application sur la forme

3.1.1.3 QPointF Application::DoForQPointF (QPointF const & *P*) const

[Application::DoForQPointF.](#)

Parameters

<i>P</i>	
----------	--

Returns

Image de l'application en P

3.1.1.4 QPointF Application::getCentre () const

[Application::getCentre.](#)

Returns

Retourne le centre de l'application

3.1.1.5 qreal Application::getK () const

[Application::getK.](#)

Returns

Retourne le rapport de l'application

3.1.1.6 qreal Application::getm11 () const

[Application::getm11.](#)

Returns

Retourne la valeur de m11

3.1.1.7 qreal Application::getm12 () const

[Application::getm12.](#)

Returns

Retourne la valeur de m12

3.1.1.8 qreal Application::getm21 () const

[Application::getm21.](#)

Returns

Retourne la valeur de m21

3.1.1.9 qreal Application::getm22 () const

[Application::getm22.](#)

Returns

Retourne la valeur de m22

3.1.1.10 qreal Application::getTheta () const

[Application::getTheta.](#)

Returns

Retourne l'angle de rotation de l'application

3.1.1.11 `qreal Application::getv1 () const`

[Application::getv1.](#)

Returns

Retourne la valeur de v1

3.1.1.12 `qreal Application::getv2 () const`

[Application::getv2.](#)

Returns

Retourne la valeur de v2

3.1.1.13 `bool Application::isHomothetie () const`

[Application::isHomothetie.](#)

Returns

Verifie si l'application est une homothetie?

3.1.1.14 `bool Application::isRotation () const`

[Application::isRotation.](#)

Returns

Verifie si l'application est une rotation?

3.1.1.15 `void Application::setCentre (QPointF P)`

[Application::setCentre.](#)

Parameters

<i>P</i>	Modifie la valeur du centre
----------	-----------------------------

3.1.1.16 `void Application::setK (qreal K)`

[Application::setK.](#)

Parameters

<i>K</i>	Modifie la valeur de k
----------	------------------------

3.1.1.17 `void Application::setm11 (qreal m)`

[Application::setm11.](#)

Parameters

<i>m</i>	Modifie la valeur de m11
----------	--------------------------

3.1.1.18 void Application::setm12 (qreal *m*)

[Application::setm12.](#)

Parameters

<i>m</i>	Modifie la valeur de m12
----------	--------------------------

3.1.1.19 void Application::setm21 (qreal *m*)

[Application::setm21.](#)

Parameters

<i>m</i>	Modifie la valeur de m21
----------	--------------------------

3.1.1.20 void Application::setm22 (qreal *m*)

[Application::setm22.](#)

Parameters

<i>m</i>	Modifie la valeur de m22
----------	--------------------------

3.1.1.21 void Application::setv1 (qreal *m*)

[Application::setv1.](#)

Parameters

<i>m</i>	Modifie la valeur de v1
----------	-------------------------

3.1.1.22 void Application::setv2 (qreal *m*)

[Application::setv2.](#)

Parameters

<i>m</i>	Modifie la valeur de v2
----------	-------------------------

The documentation for this class was generated from the following files:

- application.h
- application.cpp

3.2 Forme Class Reference

Public Member Functions

- [Forme](#) ()

- [Forme::Forme.](#)
- int [GetSize](#) () const
- [Forme::GetSize.](#)
- QPointF [GetPoint](#) (int i) const
- [Forme::GetPoint.](#)
- void [AddPoint](#) (const QPointF &P)
- [Forme::AddPoint](#) Ajoute le point P à la forme.
- void [generateExisting](#) (quint32 n=0)
- [Forme::generateExisting](#) Génère une forme par défaut
- *n=0 : Segment*
- *n=1 : Triangle.*

3.2.1 Member Function Documentation

3.2.1.1 void Forme::AddPoint (const QPointF & P)

[Forme::AddPoint](#) Ajoute le point P à la forme.

Parameters

<i>P</i>	QPointF
----------	---------

3.2.1.2 void Forme::generateExisting (quint32 n = 0)

[Forme::generateExisting](#) Génère une forme par défaut

n=0 : Segment

n=1 : Triangle.

Parameters

<i>n</i>	
----------	--

3.2.1.3 QPointF Forme::GetPoint (int i) const

[Forme::GetPoint.](#)

Parameters

<i>i</i>	Indice du point
----------	-----------------

Returns

Retourne le i-ème point(s) de la forme Nécessite que l'indice soit VALIDE.

3.2.1.4 int Forme::GetSize () const

[Forme::GetSize.](#)

Returns

Nombre de points constituant la forme

The documentation for this class was generated from the following files:

- forme.h
- forme.cpp

3.3 Fractale Class Reference

Public Member Functions

- [Fractale](#) ()
Fractale::Fractale.
- void [AddApplication](#) ([Application](#) A)
Fractale::AddApplication Ajoute une Application à la fractale.
- void [AddHomothetie](#) (qreal k)
Fractale::AddHomothetie Ajoute une [Homothetie](#) de rapport k à la fractale.
- void [AddHomothetie](#) (qreal k, [QPointF](#) Centre)
Fractale::AddHomothetie Ajoute une [Homothetie](#) de rapport k et de centre Centre à la fractale.
- void [AddHomothetie](#) (qreal k, qreal x, qreal y)
Fractale::AddHomothetie Ajoute une [Homothetie](#) de rapport k et de centre (x, y) à la fractale.
- void [AddRotation](#) (qreal theta)
Fractale::AddRotation Ajoute une [Rotation](#) d'angle theta à la fractale.
- void [AddRotation](#) (qreal theta, [QPointF](#) Centre)
Fractale::AddRotation Ajoute une [Rotation](#) d'angle theta et de centre Centre à la fractale.
- void [AddRotation](#) (qreal theta, qreal x, qreal y)
Fractale::AddRotation Ajoute une [Rotation](#) d'angle theta et de centre (x, y) à la fractale.
- void [AddForme](#) ([Forme](#) F)
Fractale::AddForme Ajoute une [Forme](#) à la fractale.
- bool [isLikeCantor](#) () const
Fractale::isLikeCantor.
- void [setLikeCantor](#) (bool p)
Fractale::setLikeCantor Attribut la valeur p à isCantor.
- void [RunOnce](#) ()
Fractale::RunOnce Calcul la fractale au rang suivant.
- [Forme](#) [getFromEnsForme](#) (int i) const
Fractale::getFromEnsForme.
- int [getSizeEnsForme](#) () const
Fractale::getSizeEnsForme.
- int [getSizeEnsAppli](#) () const
Fractale::getSizeEnsAppli.
- void [generateExisting](#) (quint32 n)
Fractale::generateExisting Génère une fractale selon des valeurs par défaut.

3.3.1 Member Function Documentation

3.3.1.1 void Fractale::AddApplication (Application A)

[Fractale::AddApplication](#) Ajoute une Application à la fractale.

Parameters

A	Application
---	-----------------------------

3.3.1.2 void Fractale::AddForme (Forme F)

[Fractale::AddForme](#) Ajoute une [Forme](#) à la fractale.

Parameters

<i>F</i>	Forme
----------	-------

3.3.1.3 void Fractale::AddHomothetie (qreal *k*)

[Fractale::AddHomothetie](#) Ajoute une [Homothetie](#) de rapport *k* à la fractale.

Parameters

<i>k</i>	Rapport de l' Homothetie
----------	--

3.3.1.4 void Fractale::AddHomothetie (qreal *k*, QPointF *Centre*)

[Fractale::AddHomothetie](#) Ajoute une [Homothetie](#) de rapport *k* et de centre *Centre* à la fractale.

Parameters

<i>k</i>	Rapport de l' Homothetie
<i>Centre</i>	Centre de l' Homothetie

3.3.1.5 void Fractale::AddHomothetie (qreal *k*, qreal *x*, qreal *y*)

[Fractale::AddHomothetie](#) Ajoute une [Homothetie](#) de rapport *k* et de centre (*x*, *y*) à la fractale.

Parameters

<i>k</i>	Rapport de l' Homothetie
<i>x</i>	Cx
<i>y</i>	Cy

3.3.1.6 void Fractale::AddRotation (qreal *theta*)

[Fractale::AddRotation](#) Ajoute une [Rotation](#) d'angle *theta* à la fractale.

Parameters

<i>theta</i>	Angle de la rotation
--------------	----------------------

3.3.1.7 void Fractale::AddRotation (qreal *theta*, QPointF *Centre*)

[Fractale::AddRotation](#) Ajoute une [Rotation](#) d'angle *theta* et de centre *Centre* à la fractale.

Parameters

<i>theta</i>	Angle de la rotation
<i>Centre</i>	Centre de la rotation

3.3.1.8 void Fractale::AddRotation (qreal *theta*, qreal *x*, qreal *y*)

[Fractale::AddRotation](#) Ajoute une [Rotation](#) d'angle *theta* et de centre (*x*, *y*) à la fractale.

Parameters

<i>theta</i>	angle de la rotation
<i>x</i>	abscisse du centre de la rotation
<i>y</i>	ordonnée du centre de la rotation

3.3.1.9 void Fractale::generateExisting (quint32 n)

[Fractale::generateExisting](#) Génère une fractale selon des valeurs par défaut.

Parameters

<i>n</i>	Type de fractal par défaut n=0 : Cantor n=2 : Triangle de Sierpinski n=3 : Courbe de Koch n=4 : Flocon de Koch n=5 : Hata's tree-like set n=6 : Lévy Curve n=7 : PentaKun n=8 : Sierpinski carpet
----------	---

3.3.1.10 Forme Fractale::getFromEnsForme (int i) const

[Fractale::getFromEnsForme](#).

Parameters

<i>i</i>	indice de la forme
----------	--------------------

Returns

Retourne la [Forme](#) d'indice i

3.3.1.11 int Fractale::getSizeEnsAppli () const

[Fractale::getSizeEnsAppli](#).

Returns

Retourne le nombre d'application définissant la fractale

3.3.1.12 int Fractale::getSizeEnsForme () const

[Fractale::getSizeEnsForme](#).

Returns

Retourne le nombre de [Forme](#)

3.3.1.13 bool Fractale::isLikeCantor () const

[Fractale::isLikeCantor](#).

Returns

Retourne si la fractale est de type Cantor

3.3.1.14 void Fractale::setLikeCantor (bool *p*)

[Fractale::setLikeCantor](#) Attribut la valeur *p* à isCantor.

Parameters

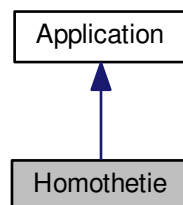
<i>p</i>	Nouvelle Valeur pour isCantor
----------	-------------------------------

The documentation for this class was generated from the following files:

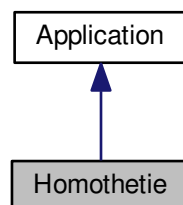
- fractale.h
- fractale.cpp

3.4 Homothetie Class Reference

Inheritance diagram for Homothetie:



Collaboration diagram for Homothetie:

**Public Member Functions**

- [Homothetie](#) ()

- [Homothetie::Homothetie](#).
- [Homothetie](#) (qreal K)
[Homothetie::Homothetie](#) Construit une [Homothetie](#) de rapport k .
- [Homothetie](#) (qreal K, QPointF C)
[Homothetie::Homothetie](#) Construit une [Homothetie](#) de rapport k et de centre C .
- void [setHomothetie](#) (qreal K)
[Homothetie::setHomothetie](#) Définit une [Homothetie](#) de rapport k .
- void [setHomothetie](#) (qreal K, QPointF C)
[Homothetie::setHomothetie](#) Définit une [Homothetie](#) de rapport k et de centre C .

Additional Inherited Members

3.4.1 Constructor & Destructor Documentation

3.4.1.1 Homothetie::Homothetie (qreal K)

[Homothetie::Homothetie](#) Construit une [Homothetie](#) de rapport k .

Parameters

K	Rapport de l' Homothetie
-----	--

3.4.1.2 Homothetie::Homothetie (qreal K, QPointF C)

[Homothetie::Homothetie](#) Construit une [Homothetie](#) de rapport k et de centre C .

Parameters

K	Rapport de l' Homothetie
C	Centre de l' Homothetie

3.4.2 Member Function Documentation

3.4.2.1 void Homothetie::setHomothetie (qreal K)

[Homothetie::setHomothetie](#) Définit une [Homothetie](#) de rapport k .

Parameters

K	Rapport de l' Homothetie
-----	--

3.4.2.2 void Homothetie::setHomothetie (qreal K, QPointF C)

[Homothetie::setHomothetie](#) Définit une [Homothetie](#) de rapport k et de centre C .

Parameters

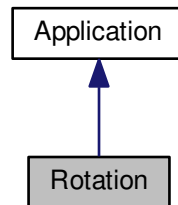
K	Rapport de l' Homothetie
C	Centre de l' Homothetie

The documentation for this class was generated from the following files:

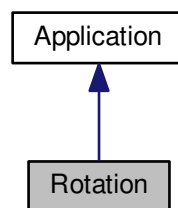
- homothetie.h
- homothetie.cpp

3.5 Rotation Class Reference

Inheritance diagram for Rotation:



Collaboration diagram for Rotation:



Public Member Functions

- [Rotation](#) ()
Rotation::Rotation.
- **Rotation** (qreal theta)
- [Rotation](#) (qreal theta, QPointF C)
Rotation::Rotation.
- void [setRotation](#) (qreal theta)
Rotation::setRotation.
- void [setRotation](#) (qreal theta, QPointF C)
Rotation::setRotation.

Additional Inherited Members

3.5.1 Constructor & Destructor Documentation

3.5.1.1 Rotation::Rotation (qreal *theta*, QPointF *C*)

[Rotation::Rotation.](#)

Parameters

<i>theta</i>	
<i>C</i>	

3.5.2 Member Function Documentation

3.5.2.1 void Rotation::setRotation (qreal *theta*)

[Rotation::setRotation.](#)

Parameters

<i>theta</i>	
--------------	--

3.5.2.2 void Rotation::setRotation (qreal *theta*, QPointF *C*)

[Rotation::setRotation.](#)

Parameters

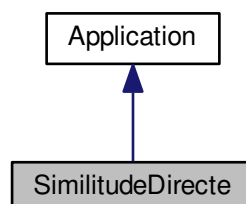
<i>theta</i>	
<i>C</i>	

The documentation for this class was generated from the following files:

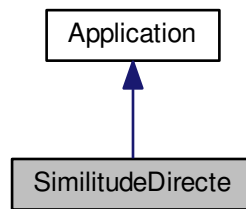
- rotation.h
- rotation.cpp

3.6 SimilitudeDirecte Class Reference

Inheritance diagram for SimilitudeDirecte:



Collaboration diagram for SimilitudeDirecte:



Public Member Functions

- [SimilitudeDirecte \(\)](#)
SimilitudeDirecte::SimilitudeDirecte.
- [SimilitudeDirecte \(qreal K, qreal theta, QPointF C\)](#)
SimilitudeDirecte::SimilitudeDirecte.
- [SimilitudeDirecte \(qreal K, qreal theta, QPointF C, QPointF P\)](#)
SimilitudeDirecte::SimilitudeDirecte.
- void [setSimilitudeDirecte \(qreal K, qreal theta, QPointF C\)](#)
SimilitudeDirecte::setSimilitudeDirecte.
- void [setSimilitudeDirecte \(qreal K, qreal theta, QPointF C, QPointF P\)](#)
SimilitudeDirecte::setSimilitudeDirecte.
- void [setTheta \(qreal theta\)](#)
SimilitudeDirecte::setTheta.

Additional Inherited Members

3.6.1 Constructor & Destructor Documentation

3.6.1.1 SimilitudeDirecte::SimilitudeDirecte (qreal K, qreal theta, QPointF C)

[SimilitudeDirecte::SimilitudeDirecte.](#)

Parameters

<i>K</i>	
<i>theta</i>	
<i>C</i>	

3.6.1.2 SimilitudeDirecte::SimilitudeDirecte (qreal K, qreal theta, QPointF C, QPointF P)

[SimilitudeDirecte::SimilitudeDirecte.](#)

Parameters

K	
θ	
C	
P	

3.6.2 Member Function Documentation

3.6.2.1 void SimilitudeDirecte::setSimilitudeDirecte (qreal K , qreal θ , QPointF C)

[SimilitudeDirecte::setSimilitudeDirecte.](#)

Parameters

K	
θ	
C	

3.6.2.2 void SimilitudeDirecte::setSimilitudeDirecte (qreal K , qreal θ , QPointF C , QPointF P)

[SimilitudeDirecte::setSimilitudeDirecte.](#)

Parameters

K	
θ	
C	
P	

3.6.2.3 void SimilitudeDirecte::setTheta (qreal θ)

[SimilitudeDirecte::setTheta.](#)

Parameters

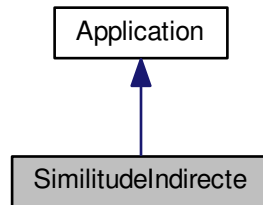
θ	
----------	--

The documentation for this class was generated from the following files:

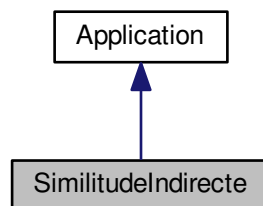
- similitudedirecte.h
- similitudedirecte.cpp

3.7 SimilitudeIndirecte Class Reference

Inheritance diagram for SimilitudeIndirecte:



Collaboration diagram for SimilitudeIndirecte:



Public Member Functions

- [SimilitudeIndirecte \(\)](#)
[SimilitudeIndirecte::SimilitudeIndirecte.](#)
- [SimilitudeIndirecte \(qreal K, qreal theta, QPointF P\)](#)
[SimilitudeIndirecte::SimilitudeIndirecte.](#)
- [SimilitudeIndirecte \(qreal K, qreal theta, QPointF P, QPointF C\)](#)
[SimilitudeIndirecte::SimilitudeIndirecte.](#)
- void [setSimilitudeIndirecte \(qreal K, qreal theta, QPointF P\)](#)
[SimilitudeIndirecte::setSimilitudeIndirecte.](#)
- void [setSimilitudeIndirecte \(qreal K, qreal theta, QPointF P, QPointF C\)](#)
[SimilitudeIndirecte::setSimilitudeIndirecte.](#)

Additional Inherited Members

3.7.1 Constructor & Destructor Documentation

3.7.1.1 `SimilitudeIndirecte::SimilitudeIndirecte (qreal K, qreal theta, QPointF P)`

[SimilitudeIndirecte::SimilitudeIndirecte.](#)

Parameters

K	
θ	
P	

3.7.1.2 SimilitudeIndirecte::SimilitudeIndirecte (qreal K , qreal θ , QPointF P , QPointF C)

[SimilitudeIndirecte::SimilitudeIndirecte.](#)

Parameters

K	
θ	
P	
C	

3.7.2 Member Function Documentation

3.7.2.1 void SimilitudeIndirecte::setSimilitudeIndirecte (qreal K , qreal θ , QPointF P)

[SimilitudeIndirecte::setSimilitudeIndirecte.](#)

Parameters

K	
θ	
P	

3.7.2.2 void SimilitudeIndirecte::setSimilitudeIndirecte (qreal K , qreal θ , QPointF P , QPointF C)

[SimilitudeIndirecte::setSimilitudeIndirecte.](#)

Parameters

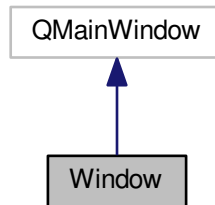
K	
θ	
P	
C	

The documentation for this class was generated from the following files:

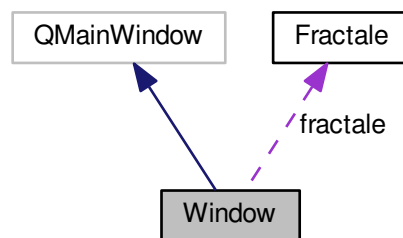
- similitudeindirecte.h
- similitudeindirecte.cpp

3.8 Window Class Reference

Inheritance diagram for Window:



Collaboration diagram for Window:



Public Slots

- void [load](#) ()
[Window::load](#) Affiche une fenetre permettant de choisir une fractale par default.
- void [refreshView](#) ()
[Window::refreshView](#) Dessine la fractale.

Public Member Functions

- [Window](#) ()
*[Window::Window](#) Constructeur par default de la classe [Window](#)
Crée les differents objets nécessaire.*

Public Attributes

- bool **tweak**
- [Fractale](#) * **fractale**

Protected Member Functions

- virtual bool [eventFilter](#) (QObject *obj, QEvent *event)
[Window::eventFilter](#) *Filtre d'événement pour capturer les evenemnts spécifique.*

3.8.1 Member Function Documentation

3.8.1.1 bool [Window::eventFilter](#) (QObject * *obj*, QEvent * *event*) [protected],[virtual]

[Window::eventFilter](#) Filtre d'événement pour capturer les evenemnts spécifique.

Parameters

<i>obj</i>	
<i>event</i>	

Returns

The documentation for this class was generated from the following files:

- window.h
- window.cpp

Index

AddApplication
 Fractale, [12](#)
AddForme
 Fractale, [12](#)
AddHomothetie
 Fractale, [13](#)
AddPoint
 Forme, [11](#)
AddRotation
 Fractale, [13](#)
Application, [5](#)
 DoForEns, [7](#)
 DoForForme, [7](#)
 DoForQPointF, [7](#)
 getCentre, [7](#)
 getK, [8](#)
 getTheta, [8](#)
 getm11, [8](#)
 getm12, [8](#)
 getm21, [8](#)
 getm22, [8](#)
 getv1, [8](#)
 getv2, [9](#)
 isHomothetie, [9](#)
 isRotation, [9](#)
 setCentre, [9](#)
 setK, [9](#)
 setm11, [9](#)
 setm12, [10](#)
 setm21, [10](#)
 setm22, [10](#)
 setv1, [10](#)
 setv2, [10](#)

DoForEns
 Application, [7](#)
DoForForme
 Application, [7](#)
DoForQPointF
 Application, [7](#)

eventFilter
 Window, [25](#)

Forme, [10](#)
 AddPoint, [11](#)
 generateExisting, [11](#)
 GetPoint, [11](#)
 GetSize, [11](#)
Fractale, [12](#)

AddApplication, [12](#)
AddForme, [12](#)
AddHomothetie, [13](#)
AddRotation, [13](#)
generateExisting, [14](#)
getFromEnsForme, [14](#)
getSizeEnsAppli, [14](#)
getSizeEnsForme, [14](#)
isLikeCantor, [14](#)
setLikeCantor, [15](#)

generateExisting
 Forme, [11](#)
 Fractale, [14](#)
getCentre
 Application, [7](#)
getFromEnsForme
 Fractale, [14](#)
getK
 Application, [8](#)
GetPoint
 Forme, [11](#)
GetSize
 Forme, [11](#)
getSizeEnsAppli
 Fractale, [14](#)
getSizeEnsForme
 Fractale, [14](#)
getTheta
 Application, [8](#)
getm11
 Application, [8](#)
getm12
 Application, [8](#)
getm21
 Application, [8](#)
getm22
 Application, [8](#)
getv1
 Application, [8](#)
getv2
 Application, [9](#)

Homothetie, [15](#)
 Homothetie, [16](#)
 setHomothetie, [16](#)

isHomothetie
 Application, [9](#)
isLikeCantor

- Fractale, [14](#)
- isRotation
 - Application, [9](#)
- Rotation, [17](#)
 - Rotation, [17](#)
 - setRotation, [18](#)
- setCentre
 - Application, [9](#)
- setHomothetie
 - Homothetie, [16](#)
- setK
 - Application, [9](#)
- setLikeCantor
 - Fractale, [15](#)
- setRotation
 - Rotation, [18](#)
- setSimilitudeDirecte
 - SimilitudeDirecte, [20](#)
- setSimilitudeIndirecte
 - SimilitudeIndirecte, [23](#)
- setTheta
 - SimilitudeDirecte, [20](#)
- setm11
 - Application, [9](#)
- setm12
 - Application, [10](#)
- setm21
 - Application, [10](#)
- setm22
 - Application, [10](#)
- setv1
 - Application, [10](#)
- setv2
 - Application, [10](#)
- SimilitudeDirecte, [18](#)
 - setSimilitudeDirecte, [20](#)
 - setTheta, [20](#)
 - SimilitudeDirecte, [19](#)
- SimilitudeIndirecte, [21](#)
 - setSimilitudeIndirecte, [23](#)
 - SimilitudeIndirecte, [21](#), [23](#)
- Window, [24](#)
 - eventFilter, [25](#)