Chaos and Fractals

1.0

Generated by Doxygen 1.8.11

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 cf::Color Struct Reference

Public Member Functions

- **Color** (uint8_t red=0, uint8_t green=0, uint8_t blue=0)
- Color **operator**∗ (float value)
- Color **operator**/ (float value)
- Color & **operator**∗**=** (float value)
- Color & **operator**/**=** (float value)
- Color **operator+** (const Color &c)
- Color **operator-** (const Color &c)
- Color & **operator+=** (const Color &c)
- Color & **operator-=** (const Color &c)
- bool **operator==** (const Color &c)
- bool **operator!=** (const Color &c)

Public Attributes

- uint8_t **b**
- uint8_t **g**
- uint8_t **r**

Static Public Attributes

- static const Color **MAGENTA**
- static const Color **YELLOW**
- static const Color **ORANGE**
- static const Color **WHITE**
- static const Color **BLACK**
- static const Color **GREEN**
- static const Color **BLUE**
- static const Color **CYAN**
- static const Color **PINK**
- static const Color **RED**

Friends

- cf::Color **operator**∗ (float value, const cf::Color &c)
- cf::Color **operator**/ (float value, const cf::Color &c)
- std::ostream & **operator**<< (std::ostream &os, const Color &c)

The documentation for this struct was generated from the following file:

- include/utils.h

2.2 cf::Intervall Struct Reference

Public Member Functions

- **Intervall** (float _min=0, float _max=0)

Static Public Member Functions

- static float **translateIntervallPostion** (const Intervall &originalIntervall, const Intervall &newIntervall, float originalPosition)

Public Attributes

- float **min**
- float **max**

Friends

- std::ostream & **operator**<< (std::ostream &os, const Intervall &intervall)

The documentation for this struct was generated from the following file:

- include/utils.h

2.3 cf::IteratedFunctionSystem Class Reference

The IteratedFunctionSystem class lazy people (like myself) may use the IFS tyepdef.

```
#include <IFS.h>
```

Public Member Functions

- void **read** (const char *fiilename)
- std::size_t **getNumTransformations** () const
- const glm::mat3x3 & **getTransformation** (std::size_t pos) const
- const Intervall & **getRangeX** () const
- const Intervall & **getRangeY** () const
- const std::string & **getName** () const
- const std::vector< glm::mat3x3 > & **getAllTransformation** () const

2.3.1 Detailed Description

The IteratedFunctionSystem class lazy people (like myself) may use the IFS tyepdef.

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

- include/IFS.h

2.4 cf::LindenmayerSystem Class Reference

The LindenmayerSystem class lazy people (like myself) may use the IFS tyepdef.

```
#include <LSystem.h>
```

Public Member Functions

- void **read** (const char *filename)
- const std::string & **getName** () const
- const std::string & **getAxiom** () const
- const std::string * **getProduction** (char symbol) const
- std::size_t **getNumProductions** () const
- bool **clearWindowEachTime** () const
- const Intervall & **getRangeX** () const
- const Intervall & **getRangeY** () const
- float **getScale** () const
- float **getStartAngle** () const
- float **getAdjustmentAngel** () const
- const std::vector< std::pair< const char, const std::string > > & **getAllProductions** () const

2.4.1 Detailed Description

The LindenmayerSystem class lazy people (like myself) may use the IFS tyepdef.

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

- include/LSystem.h

2.5 cf::Orbit Class Reference

The Orbit class lazy people (like myself) may use the ORB tyepdef.

`#include <ORB.h>`

Public Member Functions

- void **read** (const char *filename)
- const Intervall & **getRangeX** () const
- const Intervall & **getRangeY** () const
- const std::string & **getName** () const
- const std::vector< glm::vec3 > & **getAllStartingPoints** () const
- const std::vector< float > & **getAllFactors** () const
- std::size_t **getNumFactors** () const
- std::size_t **getNumStartingPoints** () const

2.5.1 Detailed Description

The Orbit class lazy people (like myself) may use the ORB tyepdef.

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

- include/ORB.h

2.6 cf::Point Struct Reference

Public Member Functions

- **Point** (float val_x, float val_y)
- bool **operator==** (const Point &p) const
- bool **operator!=** (const Point &p) const
- Point **operator+** (const Point &p) const
- Point & **operator+=** (const Point &p)
- Point **operator-** (const Point &p) const
- Point & **operator-=** (const Point &p)
- Point **operator*** (float factor) const
- Point & **operator*=** (float factor)
- Point **operator/** (float rhs) const
- Point & **operator/=** (float rhs)

Public Attributes

- float **x**
- float **y**

Friends

- Point **operator**∗ (float factor, const Point &p)
- Point **operator**/ (float lhs, const Point &p)

The documentation for this struct was generated from the following file:

- include/window2D.h

2.7 cf::Window2D Class Reference

Public Member Functions

- **Window2D** (int width=800, int height=600, const char ∗windowName="Chaos and Fractals")
- **Window2D** (const char ∗filename)
- void **show** () const
- unsigned char **waitKey** (int delay=0) const
- void **waitMouseInput** (float &x, float &y)
- void **setWindowScale** (float scale)
- float **getWindowScale** () const
- void **setInvertYAxis** (bool invert)
- bool **getInvertYAxis** () const
- void **setColor** (float x, float y, const Color &color)
- Color **getColor** (float x, float y) const
- void **drawCircle** (cf::Point point, int radius, int lineWidth, const cf::Color &color)
- void **drawRectangle** (cf::Point point1, cf::Point point2, int lineWidth, const cf::Color &color)
- void **drawLine** (cf::Point point1, cf::Point point2, int lineWidth, const cf::Color &color)
- void **setNewIntervall** (const cf::Intervall &intervallX, const cf::Intervall &intervallY)
- void **resetIntervall** ()
- void **saveImage** (const char ∗filename) const
- int **getImageWidth** () const
- int **getImageHeight** () const
- cv::Mat & **getImage** ()

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

- include/window2D.h

2.8 cf::Window3D Class Reference

Public Types

- enum **CameraType** {
  NONE, **ROTATION**, **STATIC_X_AXIS**, **STATIC_Y_AXIS**,
  STATIC_Z_AXIS }

Public Member Functions

- **Window3D** (int *argc, char **argv, int width=800, int height=600, const char *title="chaos and fractals")
- void **clear** (const Color &color=Color::BLACK)
- virtual void **draw** ()=0
- virtual void **handleKeyboardInput** (unsigned char key, int x, int y)
- int **startDrawing** ()
- int **getWindowWidth** () const
- int **getWindowHeight** () const
- void **setCamera** (CameraType type, glm::vec3 lookAt=glm::vec3(0, 0, 0), float distance=10.f)
- void **drawAxis** (float length=100.f) const
- void **drawCylinder** (const glm::vec3 &drawingDirection, const glm::vec3 &position, float diameter=1.f, const Color color=Color::WHITE) const
- void **setMaxFPS** (float maxFPS=0.f)

Static Public Member Functions

- static void **showWindowUsage** ()

Protected Attributes

- float **m_DistAdjustment** = 1.f
- float **m_AngleAdjustment** = 1.f
- float **m_CameraAdjustment** = 1.f
- glm::vec3 **m_LookAt** = glm::vec3(0.f, 0.f, 0.f)
- float **m_LookAtDistance** = 10.f

Friends

- void **_KeyboardCallbackFunction** (unsigned char key, int x, int y)
- void **_DrawingFunction** ()

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

- include/window3D.h

Index