# SIGRun

Generated by Doxygen 1.8.15

# Chapter 1

# Hierarchical Index

## 1.1 Class Hierarchy

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

# Chapter 2

# Class Index

## 2.1 Class List

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

# Chapter 3

# File Index

## 3.1 File List

Here is a list of all files with brief descriptions:

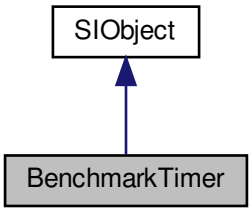
# Chapter 4

# Class Documentation

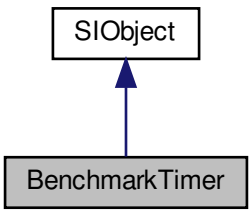
## 4.1 BenchmarkTimer Class Reference

`#include <Benchmark.hpp>`

Inheritance diagram for BenchmarkTimer:




Collaboration diagram for BenchmarkTimer:



### Public Member Functions

- BenchmarkTimer ()
- ~BenchmarkTimer ()

### Additional Inherited Members

## 4.1.1 Detailed Description

Definition at line 11 of file Benchmark.hpp.

## 4.1.2 Constructor & Destructor Documentation

#### 4.1.2.1 BenchmarkTimer()

```
BenchmarkTimer::BenchmarkTimer ( )  [inline]
```

Definition at line 14 of file Benchmark.hpp.

#### 4.1.2.2 ~BenchmarkTimer()

```
BenchmarkTimer::~BenchmarkTimer ( )  [inline]
```

Definition at line 19 of file Benchmark.hpp.

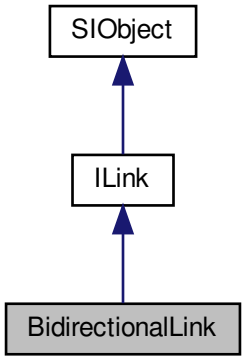
The documentation for this class was generated from the following file:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/util/Benchmark.hpp

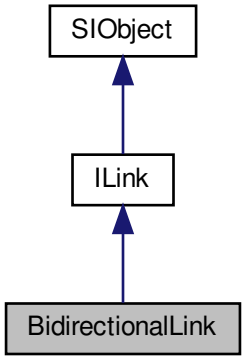
## 4.2 BidirectionalLink Class Reference

`#include <Link.hpp>`

Inheritance diagram for BidirectionalLink:




Collaboration diagram for BidirectionalLink:




### Public Member Functions

- BidirectionalLink (const std::shared_ptr< Region > &ra, const std::shared_ptr< Region > &rb, const std↩ ::string &aa, const std::string &ab)
- ~BidirectionalLink ()
- const LINK_TYPE & type () const override
- const std::shared_ptr< Region > & sender_a () const override

- const std::shared_ptr< Region > & sender_b () const override
- const std::shared_ptr< Region > & receiver_a () const override
- const std::shared_ptr< Region > & receiver_b () const override
- const std::string & attribute_a () const override
- const std::string & attribute_b () const override
- virtual void add_child (std::shared_ptr< ILink > &link) override
- std::vector< std::shared_ptr< ILink > > & children () override

### Additional Inherited Members

### 4.2.1 Detailed Description

Definition at line 70 of file Link.hpp.

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 BidirectionalLink()

```
BidirectionalLink::BidirectionalLink (
            const std::shared_ptr< Region > & ra,
            const std::shared_ptr< Region > & rb,
            const std::string & aa,
            const std::string & ab )
```

Definition at line 58 of file Link.cpp.

#### 4.2.2.2 ~BidirectionalLink()

```
BidirectionalLink::~BidirectionalLink ( )
```

Definition at line 71 of file Link.cpp.

### 4.2.3 Member Function Documentation

#### 4.2.3.1 add_child()

```
void BidirectionalLink::add_child (
            std::shared_ptr< ILink > & link )  [override], [virtual]
```

Implements ILink.

Definition at line 126 of file Link.cpp.

#### 4.2.3.2 attribute_a()

```
const std::string & BidirectionalLink::attribute_a ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 101 of file Link.cpp.

#### 4.2.3.3 attribute_b()

```
const std::string & BidirectionalLink::attribute_b ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 106 of file Link.cpp.

#### 4.2.3.4 children()

```
std::vector< std::shared_ptr< ILink > > & BidirectionalLink::children ( )  [override], [virtual]
```

Implements ILink.

Definition at line 116 of file Link.cpp.

#### 4.2.3.5 receiver_a()

```
const std::shared_ptr< Region > & BidirectionalLink::receiver_a ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 91 of file Link.cpp.

#### 4.2.3.6 receiver_b()

```
const std::shared_ptr< Region > & BidirectionalLink::receiver_b ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 96 of file Link.cpp.

#### 4.2.3.7 sender_a()

```
const std::shared_ptr< Region > & BidirectionalLink::sender_a ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 81 of file Link.cpp.

#### 4.2.3.8 sender_b()

```
const std::shared_ptr< Region > & BidirectionalLink::sender_b ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 86 of file Link.cpp.

#### 4.2.3.9 type()

```
const ILink::LINK_TYPE & BidirectionalLink::type ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 76 of file Link.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linking/Link.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linking/Link.cpp

## 4.3 Camera2D Class Reference

```
#include <Camera2D.hpp>
```

### Public Member Functions

- Camera2D ()
- ~Camera2D ()
- void initialize (int width, int height)
- void update ()
- void set_position (const glm::vec2 &position)
- const glm::vec2 & position ()
- void set_scale (float scale)
- float scale ()
- const glm::mat4 & camera_matrix () const

### 4.3.1 Detailed Description

Definition at line 9 of file Camera2D.hpp.

### 4.3.2 Constructor & Destructor Documentation

#### 4.3.2.1 Camera2D()

```
Camera2D::Camera2D ( )
```

Definition at line 5 of file Camera2D.cpp.

#### 4.3.2.2 ~Camera2D()

```
Camera2D::~Camera2D ( )
```

Definition at line 17 of file Camera2D.cpp.

### 4.3.3 Member Function Documentation

#### 4.3.3.1 camera_matrix()

```
const glm::mat4 & Camera2D::camera_matrix ( ) const
```

Definition at line 69 of file Camera2D.cpp.

#### 4.3.3.2 initialize()

```
void Camera2D::initialize (
            int width,
            int height )
```

Definition at line 22 of file Camera2D.cpp.

#### 4.3.3.3 position()

```
const glm::vec2 & Camera2D::position ( )
```

Definition at line 53 of file Camera2D.cpp.

#### 4.3.3.4 scale()

```
float Camera2D::scale ( )
```

Definition at line 64 of file Camera2D.cpp.

#### 4.3.3.5 set_position()

```
void Camera2D::set_position (
            const glm::vec2 & position )
```

Definition at line 47 of file Camera2D.cpp.

#### 4.3.3.6 set_scale()

```
void Camera2D::set_scale (
            float scale )
```

Definition at line 58 of file Camera2D.cpp.

#### 4.3.3.7 update()

```
void Camera2D::update ( )
```

Definition at line 33 of file Camera2D.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Camera2D.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Camera2D.cpp

## 4.4 Capability Class Reference

```
#include <Capability.hpp>
```

### Public Member Functions

- Capability ()
- ~Capability ()
- const std::map< std::string, int > & capabilities () const
- void add_capabilities (const bp::object &o)
- void add_capability (const std::string &name)
- void remove_capability (const std::string &name)
- int num_capabilities () const
- int consecutive_capability_id () const

### Static Public Attributes

- static std::string __test1__ = "TEST1"
- static std::string __test2__ = "TEST2"

### Friends

- class CollisionManager

## 4.4.1 Detailed Description

Definition at line 12 of file Capability.hpp.

## 4.4.2 Constructor & Destructor Documentation

### 4.4.2.1 Capability()

```
Capability::Capability ( )
```

Definition at line 10 of file Capability.cpp.

### 4.4.2.2 ~Capability()

```
Capability::~Capability ( )
```

Definition at line 17 of file Capability.cpp.

## 4.4.3 Member Function Documentation

#### 4.4.3.1 add_capabilities()

```
void Capability::add_capabilities (
            const bp::object & o )
```

Definition at line 27 of file Capability.cpp.

#### 4.4.3.2 add_capability()

```
void Capability::add_capability (
            const std::string & name )
```

Definition at line 42 of file Capability.cpp.

#### 4.4.3.3 capabilities()

```
const std::map< std::string, int > & Capability::capabilities ( ) const
```

Definition at line 22 of file Capability.cpp.

#### 4.4.3.4 consecutive_capability_id()

```
int Capability::consecutive_capability_id ( ) const
```

Definition at line 65 of file Capability.cpp.

#### 4.4.3.5 num_capabilities()

```
int Capability::num_capabilities ( ) const
```

Definition at line 60 of file Capability.cpp.

#### 4.4.3.6 remove_capability()

```
void Capability::remove_capability (
            const std::string & name )
```

Definition at line 51 of file Capability.cpp.

### 4.4.4 Friends And Related Function Documentation

#### 4.4.4.1 CollisionManager

```
friend class CollisionManager  [friend]
```

Definition at line 37 of file Capability.hpp.

### 4.4.5 Member Data Documentation

#### 4.4.5.1 __test1__

```
std::string Capability::__test1__ = "TEST1"  [static]
```

Definition at line 27 of file Capability.hpp.

#### 4.4.5.2 __test2__

```
std::string Capability::__test2__ = "TEST2"  [static]
```

Definition at line 28 of file Capability.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/Capability.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/Capability.cpp

## 4.5 CollisionManager Class Reference

```
#include <CollisionManager.hpp>
```

### Public Member Functions

- ~CollisionManager ()
- void collide (std::vector< std::shared_ptr< Region >> &regions)

### Friends

- class Context
- class RegionManager
- class SIGRunCollisionManagerTest

### 4.5.1 Detailed Description

Definition at line 12 of file CollisionManager.hpp.

### 4.5.2 Constructor & Destructor Documentation

#### 4.5.2.1 ~CollisionManager()

```
CollisionManager::~CollisionManager ( )
```

Definition at line 8 of file CollisionManager.cpp.

### 4.5.3 Member Function Documentation

#### 4.5.3.1 collide()

```
void CollisionManager::collide (
            std::vector< std::shared_ptr< Region >> & regions )
```

Definition at line 13 of file CollisionManager.cpp.

### 4.5.4 Friends And Related Function Documentation

#### 4.5.4.1 Context

```
friend class Context  [friend]
```

Definition at line 54 of file CollisionManager.hpp.

#### 4.5.4.2 RegionManager

```
friend class RegionManager  [friend]
```

Definition at line 55 of file CollisionManager.hpp.

#### 4.5.4.3 SIGRunCollisionManagerTest

```
friend class SIGRunCollisionManagerTest  [friend]
```

Definition at line 56 of file CollisionManager.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/CollisionManager.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/CollisionManager.cpp

## 4.6 Color Struct Reference

```
#include <Vertex.hpp>
```

### Public Member Functions

- Color ()
- Color (GLubyte R, GLubyte G, GLubyte B, GLubyte A)

### Public Attributes

- GLubyte r
- GLubyte g
- GLubyte b
- GLubyte a

### 4.6.1 Detailed Description

Definition at line 16 of file Vertex.hpp.

### 4.6.2 Constructor & Destructor Documentation

#### 4.6.2.1 Color() [1/2]

```
Color::Color ( )  [inline]
```

Definition at line 18 of file Vertex.hpp.

#### 4.6.2.2 Color() [2/2]

```
Color::Color (
            GLubyte R,
            GLubyte G,
            GLubyte B,
            GLubyte A )  [inline]
```

Definition at line 19 of file Vertex.hpp.

### 4.6.3 Member Data Documentation

#### 4.6.3.1 a

```
GLubyte Color::a
```

Definition at line 26 of file Vertex.hpp.

#### 4.6.3.2 b

```
GLubyte Color::b
```

Definition at line 25 of file Vertex.hpp.

#### 4.6.3.3 g

```
GLubyte Color::g
```

Definition at line 24 of file Vertex.hpp.

#### 4.6.3.4 r

`GLubyte Color::r`

Definition at line 23 of file Vertex.hpp.

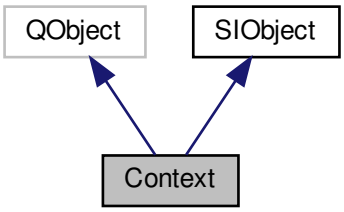
The documentation for this struct was generated from the following file:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Vertex.hpp

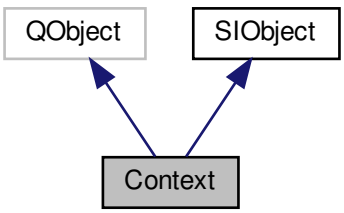
## 4.7 Context Class Reference

`#include <Context.hpp>`

Inheritance diagram for Context:




Collaboration diagram for Context:



### Public Member Functions

- ~Context ()
- void begin (IRenderEngine *ire, int argc, char **argv)
- Capability * capability_manager ()
- RegionManager * region_manager ()
- void update ()
- void enable (int what)
- void disable (int what)
- int width ()
- int height ()

### Static Public Member Functions

- static Context * SIContext ()

### Friends

- class Core

### Additional Inherited Members

## 4.7.1 Detailed Description

Definition at line 18 of file Context.hpp.

## 4.7.2 Constructor & Destructor Documentation

#### 4.7.2.1 ~Context()

```
Context::~Context ( )
```

Definition at line 13 of file Context.cpp.

## 4.7.3 Member Function Documentation

#### 4.7.3.1 begin()

```
void Context::begin (
            IRenderEngine * ire,
            int argc,
            char ** argv )
```

Definition at line 43 of file Context.cpp.

#### 4.7.3.2 capability_manager()

```
Capability * Context::capability_manager ( )
```

Definition at line 83 of file Context.cpp.

#### 4.7.3.3 disable()

```
void Context::disable (
            int what )
```

Definition at line 119 of file Context.cpp.

#### 4.7.3.4 enable()

```
void Context::enable (
            int what )
```

Definition at line 114 of file Context.cpp.

#### 4.7.3.5 height()

```
int Context::height ( )
```

Definition at line 109 of file Context.cpp.

#### 4.7.3.6 region_manager()

```
RegionManager * Context::region_manager ( )
```

Definition at line 88 of file Context.cpp.

#### 4.7.3.7 SIContext()

```
Context * Context::SIContext ( )  [static]
```

Definition at line 93 of file Context.cpp.

#### 4.7.3.8 update()

```
void Context::update ( )
```

Definition at line 98 of file Context.cpp.

#### 4.7.3.9 width()

```
int Context::width ( )
```

Definition at line 104 of file Context.cpp.

### 4.7.4 Friends And Related Function Documentation

#### 4.7.4.1 Core

```
friend class Core  [friend]
```

Definition at line 53 of file Context.hpp.

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

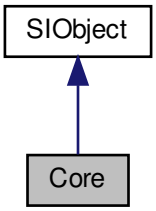
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/Context.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/Context.cpp

## 4.8 Core Class Reference

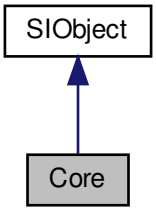
namespace shortening for python object integration

`#include <Core.hpp>`

Inheritance diagram for Core:




Collaboration diagram for Core:




### Public Member Functions

- ~Core ()
  *destructor*
- void start (char **argv, int argc, IRenderEngine *ire)
  *entry point of core SIGRun initialization*
- void stop ()
  *exit SIGRun core*

### Protected Member Functions

- Core ()
  *constructor*
- void retrieve_available_plugins (std::unordered_map< std::string, std::unique_ptr< bp::object >> &plugins, const std::string &plugin_path)
  *retrieve all available plugins before launching SIGRun environment*

### Friends

- class SIGRun
- class SIGRunTest
- class SIGRunCoreTest

### Additional Inherited Members

## 4.8.1 Detailed Description

namespace shortening for python object integration

SIObject Central Core class registered as SIObject

This class initiates all subsystems required for the SIGRun environment. This class collects all available pulgins first. Second, it launches the SI context and other subsystems. This class is registered as SIObject meta type. This class conctructor is declared private to disable use by external application programmers. Therefore, the friend keyword is used to internally expose the class.

Definition at line 29 of file Core.hpp.

## 4.8.2 Constructor & Destructor Documentation

### 4.8.2.1 ~Core()

```
Core::~Core ( )
```

destructor

Shut down the SIGRun environment.

Definition at line 27 of file Core.cpp.

### 4.8.2.2 Core()

```
Core::Core ( )  [protected]
```

constructor

Constructor which registers instance as an SIObject. Specify, which Logging capabilities are desired.

Definition at line 15 of file Core.cpp.

## 4.8.3 Member Function Documentation

### 4.8.3.1 retrieve_available_plugins()

```
void Core::retrieve_available_plugins (
            std::unordered_map< std::string, std::unique_ptr< bp::object >> & plugins,
            const std::string & plugin_path )  [protected]
```

retrieve all available plugins before launching SIGRun environment

Load all plugins in the plugin path of the SIGRun environment.

**Parameters**

| | |
|---|---|
| *plugins* | a mutable reference to a std::unordered map with std::string as key and a std::shared_ptr of boost::python::objects as values which is the out parameter |
| *plugin_path* | a std::string which contains the path to the root folder of all plugin files |

**See also**

Scripting::Scripting
PluginCollector::PluginCollector

Definition at line 85 of file Core.cpp.

#### 4.8.3.2 start()

```
void Core::start (
            char ** argv,
            int argc,
            IRenderEngine * ire )
```

entry point of core SIGRun initialization

Entry point of SIGRun's core which performs Plugin loading and initializes the SI Context.

Definition at line 39 of file Core.cpp.

#### 4.8.3.3 stop()

```
void Core::stop ( )
```

exit SIGRun core

Initiate the shutdown of the SIGRun core.

Definition at line 69 of file Core.cpp.

### 4.8.4 Friends And Related Function Documentation

#### 4.8.4.1 SIGRun

```
friend class SIGRun  [friend]
```

Definition at line 42 of file Core.hpp.

#### 4.8.4.2 SIGRunCoreTest

```
friend class SIGRunCoreTest  [friend]
```

Definition at line 44 of file Core.hpp.

#### 4.8.4.3 SIGRunTest

```
friend class SIGRunTest  [friend]
```

Definition at line 43 of file Core.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/Core.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/Core.cpp

## 4.9 GLSLProgram Class Reference

```
#include <GLSLProgram.hpp>
```

### Public Member Functions

- GLSLProgram ()
- ~GLSLProgram ()
- void use ()
- void unuse ()
- GLint uniform_location (const std::string &uniform_name)
- void compile_shaders (const std::string &vertex_shader_filepath, const std::string &fragment_shader_↩ filepath)
- void link_shaders ()
- void add_attribute (const std::string &attribute_name)

### 4.9.1 Detailed Description

Definition at line 9 of file GLSLProgram.hpp.

### 4.9.2 Constructor & Destructor Documentation

#### 4.9.2.1 GLSLProgram()

```
GLSLProgram::GLSLProgram ( )
```

Definition at line 6 of file GLSLProgram.cpp.

#### 4.9.2.2 ~GLSLProgram()

```
GLSLProgram::~GLSLProgram ( )
```

Definition at line 13 of file GLSLProgram.cpp.

### 4.9.3 Member Function Documentation

#### 4.9.3.1 add_attribute()

```
void GLSLProgram::add_attribute (
            const std::string & attribute_name )
```

Definition at line 98 of file GLSLProgram.cpp.

#### 4.9.3.2 compile_shaders()

```
void GLSLProgram::compile_shaders (
            const std::string & vertex_shader_filepath,
            const std::string & fragment_shader_filepath )
```

Definition at line 45 of file GLSLProgram.cpp.

#### 4.9.3.3 link_shaders()

```
void GLSLProgram::link_shaders ( )
```

Definition at line 69 of file GLSLProgram.cpp.

#### 4.9.3.4 uniform_location()

```
GLint GLSLProgram::uniform_location (
            const std::string & uniform_name )
```

Definition at line 32 of file GLSLProgram.cpp.

#### 4.9.3.5 unuse()

```
void GLSLProgram::unuse ( )
```

Definition at line 24 of file GLSLProgram.cpp.

#### 4.9.3.6 use()

```
void GLSLProgram::use ( )
```

Definition at line 16 of file GLSLProgram.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/shader/GLSLProgram.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/shader/GLSLProgram.cpp

## 4.10 GLTexture Struct Reference

`#include <GLTexture.hpp>`

### Public Attributes

- GLuint id
- int width
- int height

### 4.10.1 Detailed Description

Definition at line 7 of file GLTexture.hpp.

### 4.10.2 Member Data Documentation

#### 4.10.2.1 height

```
int GLTexture::height
```

Definition at line 11 of file GLTexture.hpp.

#### 4.10.2.2 id

```
GLuint GLTexture::id
```

Definition at line 9 of file GLTexture.hpp.

#### 4.10.2.3 width

```
int GLTexture::width
```

Definition at line 10 of file GLTexture.hpp.

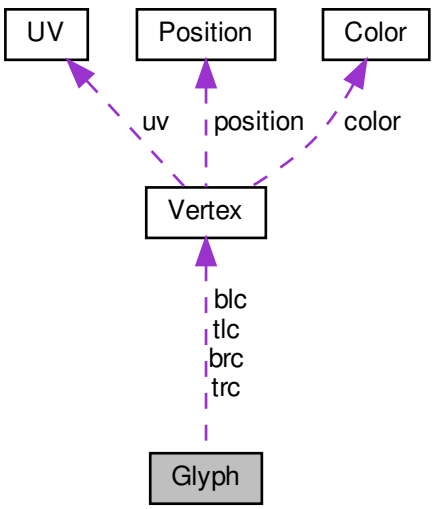
The documentation for this struct was generated from the following file:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/GLTexture.hpp

## 4.11 Glyph Class Reference

```
#include <Glyph.hpp>
```

Collaboration diagram for Glyph:



### Public Member Functions

- Glyph ()
- Glyph (const RegionRepresentation *region)
- Glyph (const glm::vec4 &destination_rect, const glm::vec4 &uv_rect, const std::string &tex, float d, const Color &color)

### Public Attributes

- GLuint texture
- float depth
- glm::mat4 transform
- std::vector< Vertex > vertices
- Vertex tlc
- Vertex blc
- Vertex brc
- Vertex trc

## 4.11.1 Detailed Description

Definition at line 21 of file Glyph.hpp.

## 4.11.2 Constructor & Destructor Documentation

### 4.11.2.1 Glyph() [1/3]

```
Glyph::Glyph ( )  [inline]
```

Definition at line 24 of file Glyph.hpp.

### 4.11.2.2 Glyph() [2/3]

```
Glyph::Glyph (
            const RegionRepresentation * region )  [inline]
```

Definition at line 27 of file Glyph.hpp.

#### 4.11.2.3 Glyph() [3/3]

```
Glyph::Glyph (
            const glm::vec4 & destination_rect,
            const glm::vec4 & uv_rect,
            const std::string & tex,
            float d,
            const Color & color )  [inline]
```

Definition at line 54 of file Glyph.hpp.

### 4.11.3 Member Data Documentation

#### 4.11.3.1 blc

```
Vertex Glyph::blc
```

Definition at line 80 of file Glyph.hpp.

#### 4.11.3.2 brc

```
Vertex Glyph::brc
```

Definition at line 80 of file Glyph.hpp.

#### 4.11.3.3 depth

```
float Glyph::depth
```

Definition at line 75 of file Glyph.hpp.

#### 4.11.3.4 texture

```
GLuint Glyph::texture
```

Definition at line 74 of file Glyph.hpp.

#### 4.11.3.5 tlc

```
Vertex Glyph::tlc
```

Definition at line 80 of file Glyph.hpp.

#### 4.11.3.6 transform

```
glm::mat4 Glyph::transform
```

Definition at line 76 of file Glyph.hpp.

#### 4.11.3.7 trc

```
Vertex Glyph::trc
```

Definition at line 80 of file Glyph.hpp.

#### 4.11.3.8 vertices

```
std::vector<Vertex> Glyph::vertices
```

Definition at line 78 of file Glyph.hpp.

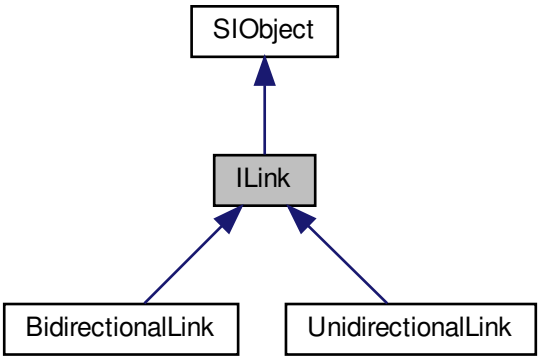
The documentation for this class was generated from the following file:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Glyph.hpp

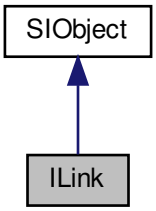
## 4.12 ILink Class Reference

`#include <Link.hpp>`

Inheritance diagram for ILink:




Collaboration diagram for ILink:




### Public Types

- enum LINK_TYPE { UD, BD }

### Public Member Functions

- virtual ~ILink ()
- virtual const LINK_TYPE & type () const =0
- virtual const std::shared_ptr< Region > & sender_a () const =0
- virtual const std::shared_ptr< Region > & sender_b () const =0
- virtual const std::shared_ptr< Region > & receiver_a () const =0
- virtual const std::shared_ptr< Region > & receiver_b () const =0
- virtual const std::string & attribute_a () const =0
- virtual const std::string & attribute_b () const =0
- virtual void add_child (std::shared_ptr< ILink > &link)=0
- virtual std::vector< std::shared_ptr< ILink > > & children ()=0

### Additional Inherited Members

### 4.12.1 Detailed Description

Definition at line 15 of file Link.hpp.

### 4.12.2 Member Enumeration Documentation

#### 4.12.2.1 LINK_TYPE

```
enum ILink::LINK_TYPE
```

**Enumerator**

| | |
|---|---|
| UD | |
| BD | |

Definition at line 18 of file Link.hpp.

### 4.12.3 Constructor & Destructor Documentation

#### 4.12.3.1 ~ILink()

```
virtual ILink::~ILink ( )  [inline], [virtual]
```

Definition at line 24 of file Link.hpp.

### 4.12.4 Member Function Documentation

#### 4.12.4.1 add_child()

```
virtual void ILink::add_child (
            std::shared_ptr< ILink > & link )  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

#### 4.12.4.2 attribute_a()

```
virtual const std::string& ILink::attribute_a ( ) const  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

#### 4.12.4.3 attribute_b()

```
virtual const std::string& ILink::attribute_b ( ) const  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

#### 4.12.4.4 children()

```
virtual std::vector<std::shared_ptr<ILink> >& ILink::children ( )  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

#### 4.12.4.5 receiver_a()

```
virtual const std::shared_ptr<Region>& ILink::receiver_a ( ) const  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

#### 4.12.4.6 receiver_b()

```
virtual const std::shared_ptr<Region>& ILink::receiver_b ( ) const  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

#### 4.12.4.7 sender_a()

```
virtual const std::shared_ptr<Region>& ILink::sender_a ( ) const  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

#### 4.12.4.8 sender_b()

```
virtual const std::shared_ptr<Region>& ILink::sender_b ( ) const  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

#### 4.12.4.9 type()

```
virtual const LINK_TYPE& ILink::type ( ) const  [pure virtual]
```

Implemented in BidirectionalLink, and UnidirectionalLink.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linking/Link.hpp

## 4.13 ImageLoader Class Reference

```
#include <ImageLoader.hpp>
```

### Static Public Member Functions

- static GLTexture load_png (const char *filepath)

### 4.13.1 Detailed Description

Definition at line 8 of file ImageLoader.hpp.

### 4.13.2 Member Function Documentation

#### 4.13.2.1 load_png()

```
GLTexture ImageLoader::load_png (
            const char * filepath )  [static]
```

Definition at line 9 of file ImageLoader.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/ImageLoader.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/ImageLoader.cpp

## 4.14 IOManager Class Reference

`#include <IOManager.hpp>`

### Static Public Member Functions

- static bool read_file_to_buffer (const std::string &filepath, std::vector< char > &buffer)

### 4.14.1 Detailed Description

Definition at line 9 of file IOManager.hpp.

### 4.14.2 Member Function Documentation

#### 4.14.2.1 read_file_to_buffer()

```
bool IOManager::read_file_to_buffer (
            const std::string & filepath,
            std::vector< char > & buffer )  [static]
```

Definition at line 6 of file IOManager.cpp.

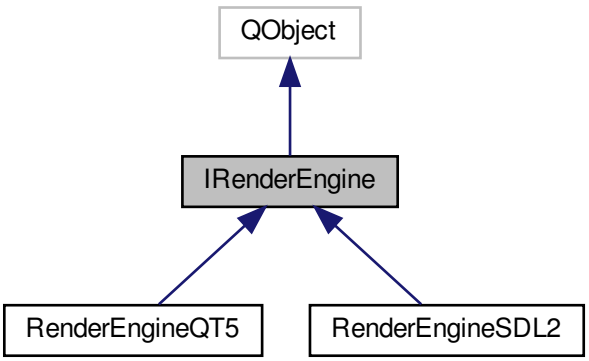
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/IOManager.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/IOManager.cpp

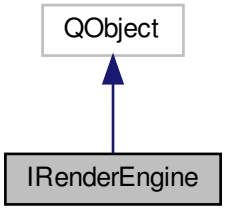
## 4.15 IRenderEngine Class Reference

`#include <IRenderEngine.hpp>`

Inheritance diagram for IRenderEngine:



Collaboration diagram for IRenderEngine:




### Public Member Functions

- virtual void start (int width, int height)=0
- virtual void run ()=0
- virtual void pause ()=0

## 4.15.1 Detailed Description

Definition at line 11 of file IRenderEngine.hpp.

## 4.15.2 Member Function Documentation

#### 4.15.2.1 pause()

```
virtual void IRenderEngine::pause ( )  [pure virtual]
```

Implemented in RenderEngineSDL2, and RenderEngineQT5.

#### 4.15.2.2 run()

```
virtual void IRenderEngine::run ( )  [pure virtual]
```

Implemented in RenderEngineSDL2, and RenderEngineQT5.

#### 4.15.2.3 start()

```
virtual void IRenderEngine::start (
            int width,
            int height )  [pure virtual]
```

Implemented in RenderEngineQT5.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/rendering/IRenderEngine.hpp

## 4.16 IterableConverter Class Reference

```
#include <SuperEffect.hpp>
```

### Public Member Functions

- template<typename Container >
  IterableConverter & from_python ()

### Static Public Member Functions

- static void * convertible (PyObject *object)
  *Check if PyObject is iterable.*
- template<typename Container >
  static void construct (PyObject *object, bp::converter::rvalue_from_python_stage1_data *data)
  *Convert iterable PyObject to C++ container type.*

### 4.16.1 Detailed Description

Definition at line 10 of file SuperEffect.hpp.

### 4.16.2 Member Function Documentation

#### 4.16.2.1 construct()

```
template<typename Container >
void IterableConverter::construct (
            PyObject * object,
            bp::converter::rvalue_from_python_stage1_data * data )  [static]
```

Convert iterable PyObject to C++ container type.

Container Concept requirements:

- Container::value_type is CopyConstructable.
- Container can be constructed and populated with two iterators. I.e. Container(begin, end)

Definition at line 24 of file SuperEffect.cpp.

#### 4.16.2.2 convertible()

```
void * IterableConverter::convertible (
            PyObject * object )  [static]
```

Check if PyObject is iterable.

Definition at line 18 of file SuperEffect.cpp.

#### 4.16.2.3 from_python()

```
template<typename Container >
IterableConverter & IterableConverter::from_python ( )
```

**Note**

Registers converter from a python interable type to the provided type.

Definition at line 10 of file SuperEffect.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/pysi/SuperEffect.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/pysi/SuperEffect.cpp

## 4.17 LinkingGraph Class Reference

```
#include <LinkingGraph.hpp>
```

### Public Member Functions

- LinkingGraph ()
- ~LinkingGraph ()
- void add_link (std::shared_ptr< Region > &a, const std::string &attr_a, std::shared_ptr< Region > &b, const std::string &attr_b, const ILink::LINK_TYPE &link_type)
- void remove_link (std::shared_ptr< Region > &a, const std::string &attr_a, std::shared_ptr< Region > &b, const std::string &attr_b, const ILink::LINK_TYPE &link_type)
- void emit_link_event (std::shared_ptr< Region > &a, const std::string &attr_a)
- bool is_linked (const std::shared_ptr< Region > &a, const std::string &attr_a, const std::shared_ptr< Region > &b, const std::string &attr_b, const ILink::LINK_TYPE &link_type) const
- const std::vector< std::shared_ptr< ILink > > & links () const

### 4.17.1 Detailed Description

Definition at line 12 of file LinkingGraph.hpp.

### 4.17.2 Constructor & Destructor Documentation

#### 4.17.2.1 LinkingGraph()

```
LinkingGraph::LinkingGraph ( )
```

Definition at line 6 of file LinkingGraph.cpp.

#### 4.17.2.2 ~LinkingGraph()

```
LinkingGraph::~LinkingGraph ( )
```

Definition at line 11 of file LinkingGraph.cpp.

### 4.17.3 Member Function Documentation

#### 4.17.3.1 add_link()

```
void LinkingGraph::add_link (
            std::shared_ptr< Region > & a,
            const std::string & attr_a,
            std::shared_ptr< Region > & b,
            const std::string & attr_b,
            const ILink::LINK_TYPE & link_type )
```

Definition at line 16 of file LinkingGraph.cpp.

#### 4.17.3.2 emit_link_event()

```
void LinkingGraph::emit_link_event (
            std::shared_ptr< Region > & a,
            const std::string & attr_a )
```

Definition at line 84 of file LinkingGraph.cpp.

#### 4.17.3.3 is_linked()

```
bool LinkingGraph::is_linked (
            const std::shared_ptr< Region > & a,
            const std::string & attr_a,
            const std::shared_ptr< Region > & b,
            const std::string & attr_b,
            const ILink::LINK_TYPE & link_type ) const
```

Definition at line 96 of file LinkingGraph.cpp.

#### 4.17.3.4 links()

```
const std::vector< std::shared_ptr< ILink > > & LinkingGraph::links ( ) const
```

Definition at line 140 of file LinkingGraph.cpp.

#### 4.17.3.5 remove_link()

```
void LinkingGraph::remove_link (
            std::shared_ptr< Region > & a,
            const std::string & attr_a,
            std::shared_ptr< Region > & b,
            const std::string & attr_b,
            const ILink::LINK_TYPE & link_type )
```

Definition at line 33 of file LinkingGraph.cpp.

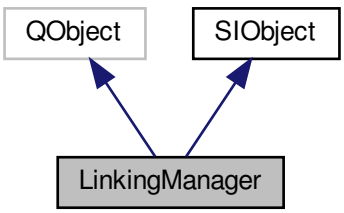
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linking/LinkingGraph.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linking/LinkingGraph.cpp

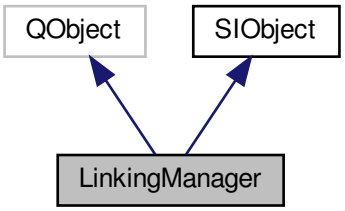
## 4.18 LinkingManager Class Reference

```
#include <LinkingManager.hpp>
```

Inheritance diagram for LinkingManager:



Collaboration diagram for LinkingManager:




## Public Member Functions

- LinkingManager ()
- ~LinkingManager ()
- bool add_link (std::shared_ptr< Region > &ra, const std::string &aa, std::shared_ptr< Region > &rb, const std::string &ab, const ILink::LINK_TYPE &type)
- void remove_link (std::shared_ptr< Region > &ra, const std::string &aa, std::shared_ptr< Region > &rb, const std::string &ab, const ILink::LINK_TYPE &type)
- bool is_linked (const std::shared_ptr< Region > &ra, const std::string &aa, const std::shared_ptr< Region > &rb, const std::string &ab, const ILink::LINK_TYPE &type)
- void emit_link_event (std::shared_ptr< Region > &a, const std::string &attr_a)
- const std::vector< std::shared_ptr< ILink > > & links () const
- const std::unique_ptr< LinkingGraph > & linking_graph () const
- const int num_links () const

## Additional Inherited Members

### 4.18.1 Detailed Description

Definition at line 20 of file LinkingManager.hpp.

### 4.18.2 Constructor & Destructor Documentation

#### 4.18.2.1 LinkingManager()

```
LinkingManager::LinkingManager ( )
```

Definition at line 7 of file LinkingManager.cpp.

#### 4.18.2.2 ~LinkingManager()

```
LinkingManager::~LinkingManager ( )
```

Definition at line 15 of file LinkingManager.cpp.

### 4.18.3 Member Function Documentation

#### 4.18.3.1 add_link()

```
bool LinkingManager::add_link (
            std::shared_ptr< Region > & ra,
            const std::string & aa,
            std::shared_ptr< Region > & rb,
            const std::string & ab,
            const ILink::LINK_TYPE & type )
```

Definition at line 22 of file LinkingManager.cpp.

#### 4.18.3.2 emit_link_event()

```
void LinkingManager::emit_link_event (
            std::shared_ptr< Region > & a,
            const std::string & attr_a )
```

Definition at line 125 of file LinkingManager.cpp.

#### 4.18.3.3 is_linked()

```
bool LinkingManager::is_linked (
            const std::shared_ptr< Region > & ra,
            const std::string & aa,
            const std::shared_ptr< Region > & rb,
            const std::string & ab,
            const ILink::LINK_TYPE & type )
```

Definition at line 119 of file LinkingManager.cpp.

#### 4.18.3.4 linking_graph()

```
const std::unique_ptr< LinkingGraph > & LinkingManager::linking_graph ( ) const
```

Definition at line 135 of file LinkingManager.cpp.

#### 4.18.3.5 links()

```
const std::vector< std::shared_ptr< ILink > > & LinkingManager::links ( ) const
```

Definition at line 130 of file LinkingManager.cpp.

#### 4.18.3.6 num_links()

```
const int LinkingManager::num_links ( ) const
```

Definition at line 140 of file LinkingManager.cpp.

#### 4.18.3.7 remove_link()

```
void LinkingManager::remove_link (
            std::shared_ptr< Region > & ra,
            const std::string & aa,
            std::shared_ptr< Region > & rb,
            const std::string & ab,
            const ILink::LINK_TYPE & type )
```

Definition at line 100 of file LinkingManager.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/LinkingManager.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/LinkingManager.cpp

## 4.19 Log Class Reference

Log class serving as central logging functionality for easy logging data output.

```
#include <Log.hpp>
```

### Public Types

- enum LOG_LEVEL {
  INFO_LEVEL = 0b00001, WARN_LEVEL = 0b00010, DEBUG_LEVEL = 0b00100, ERROR_LEVEL = 0b01000,
  UNDEFINED_LEVEL = 0b10000 }

  *enum for log level selection modelled as a bitfield*
- enum MODE { NONE = 0, CONSOLE = 1, FILE = 2 }

  *enum for log mode selection modelled as a bitfield*
- enum SHOW_TYPE {
  HIDDEN = 0, INFO = 1, WARN = 2, DEBUG = 4,
  ERROR = 8, UNDEFINED = 16 }

  *enum for log show type selection modelled as a bitfield*

### Static Public Member Functions

- static void log (const std::string &origin, const std::string &what, int level, const std::string &type, const std↩ ::string &file="", const std::string &func="", const std::string &line="")

  *central logging function outputting log messages according to its params*
- static void set_log_file_path (const std::string &path)
- static std::string log_level (int log_level)

  *return the level of a log message as tag according to its id*
- static std::string time ()

  *return current system time with milliseconds precision*

### Static Public Attributes

- static std::string log_file_path = Log::PATH_DEFAULT
- static int SHOW = -1

  *the integer variable containing which log messages are outputted based on their tag*
- static int WHERE = 0
- static bool __DEBUG__ = false

  *the flag which is required to be set to true if the logging system is required to be used.*

#### 4.19.1 Detailed Description

Log class serving as central logging functionality for easy logging data output.

This class serves as the central knot for all output operations in terms of log messages. This class is a static class featuring no ctor or dtor. Shortcut macros make the access to this class logging functionality more easier.

**See also**

DEBUG(what, log_mode)
WARN(what, log_mode)
ERROR(what, log_mode)
INFO(what, log_mode)
UNDEFINED(what, log_mode)

Definition at line 181 of file Log.hpp.

### 4.19.2 Member Enumeration Documentation

#### 4.19.2.1 LOG_LEVEL

`enum Log::LOG_LEVEL`

enum for log level selection modelled as a bitfield

The log level describes which tag is assigned to a log message.

**Enumerator**

| | |
|---|---|
| INFO_LEVEL | |
| WARN_LEVEL | |
| DEBUG_LEVEL | |
| ERROR_LEVEL | |
| UNDEFINED_LEVEL | |

Definition at line 206 of file Log.hpp.

#### 4.19.2.2 MODE

`enum Log::MODE`

enum for log mode selection modelled as a bitfield

The log mode describes where a log message is outputted. A mode is ignored if it is not specified. Due to the enum being modelled as a bitfield, users can use the | operator to selectively enable modes for logging output. Example for enabling printing to stdout as well as to a file: int mode = CONSOLE | FILE;

**Enumerator**

| | |
|---|---|
| NONE | |
| CONSOLE | |
| FILE | |

Definition at line 224 of file Log.hpp.

#### 4.19.2.3 SHOW_TYPE

`enum Log::SHOW_TYPE`

enum for log show type selection modelled as a bitfield

The log show type describes which log messages are outputted based on their tags. Tags which are not specified are ignored. Due to the enum being modelled as a bitfield, users can use the | operator to selectively enable tags for logging output. Example for enabling DEBUG and WARN tags without the INFO tag: int loglevel = WARN | DEBUG

**Enumerator**

| | |
|---|---|
| HIDDEN | |
| INFO | |
| WARN | |
| DEBUG | |
| ERROR | |
| UNDEFINED | |

Definition at line 239 of file Log.hpp.

### 4.19.3 Member Function Documentation

#### 4.19.3.1 log()

```
void Log::log (
            const std::string & origin,
            const std::string & what,
            int level,
            const std::string & type,
            const std::string & file = "",
            const std::string & func = "",
            const std::string & line = "" )  [static]
```

central logging function outputting log messages according to its params

This is the central logging function of SIGRun. It requires to be called from class which are registered as SIObject. The parameters of this function, besides what (log message), configure the way the message is outputted. This static method is easier accessible via the shortcut macros.

**Parameters**

| | |
|---|---|
| *what* | a std::string containing the log message |
| *level* | an integer containing the id of the desired tag |
| *logging_flags* | an integer containing where the log message is to be outputted |
| *type* | a std::string containing the description of the functions caller via an SIObject |
| *file* | a std::string containing the name of the file in which the log call is implemented |
| *func* | a std::string containing the name of the function in which the log call was issued |
| *line* | a std::string containing the number of the line of the file in which the log call is implemented |

**See also**

DEBUG(what, log_mode)
WARN(what, log_mode)

ERROR(what, log_mode)
INFO(what, log_mode)
UNDEFINED(what, log_mode)
SIObject

Definition at line 38 of file Log.cpp.

#### 4.19.3.2 log_level()

```
std::string Log::log_level (
            int log_level )  [static]
```

return the level of a log message as tag according to its id

Retrieves the level of a log message according to the value of the parameter which is compared to the Log::LOG_LEVEL enum/bitfield.

**Parameters**

| | |
|---|---|
| *log_level* | an integer containing the id of the desired tag |

**Returns**

a std::string which contains a human readable version of the desired tag

Definition at line 113 of file Log.cpp.

#### 4.19.3.3 set_log_file_path()

```
void Log::set_log_file_path (
            const std::string & path )  [static]
```

set the path of the file for logging output Set the value of the static variable log_file_path to the value of the given parameter to specifiy the file path of the log output.

**Parameters**

| | |
|---|---|
| *path* | a std::string containing the desired file path for logging to files |

Definition at line 99 of file Log.cpp.

#### 4.19.3.4 time()

```
std::string Log::time ( )  [static]
```

return current system time with milliseconds precision

Compute current system time with milliseconds precision. Format the date data to yyyy-MM-dd hh:mm↩ :ss.<milliseconds>. Concatenate the date data to a std::string.

**Returns**

a std::string containing the formatted date data

Definition at line 139 of file Log.cpp.

### 4.19.4 Member Data Documentation

#### 4.19.4.1 __DEBUG__

```
bool Log::__DEBUG__ = false  [static]
```

the flag which is required to be set to true if the logging system is required to be used.

This flag is the center of enabling (**DEBUG** is set to true) or disabling (**DEBUG** is set to false) the entire logging system.

Definition at line 266 of file Log.hpp.

#### 4.19.4.2 log_file_path

```
std::string Log::log_file_path = Log::PATH_DEFAULT  [static]
```

actual path to logfile

Definition at line 199 of file Log.hpp.

#### 4.19.4.3 SHOW

```
int Log::SHOW = -1  [static]
```

the integer variable containing which log messages are outputted based on their tag

This integer variable regulates which log messages are outputted, according to their tags.

Definition at line 254 of file Log.hpp.

#### 4.19.4.4 WHERE

```
int Log::WHERE = 0  [static]
```

Definition at line 259 of file Log.hpp.

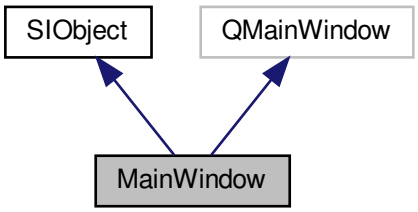
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/log/Log.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/log/Log.cpp

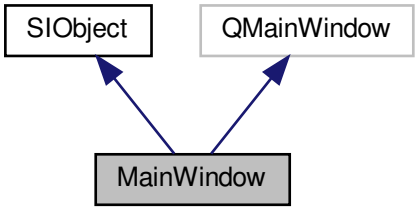
## 4.20 MainWindow Class Reference

```
#include <MainWindow.hpp>
```

Inheritance diagram for MainWindow:




Collaboration diagram for MainWindow:




### Public Member Functions

- MainWindow (int width, int height)
- ~MainWindow ()
- void set_is_running (bool running)

### Protected Member Functions

- void paintEvent (QPaintEvent ∗event) override
- void keyPressEvent (QKeyEvent ∗event) override

### Additional Inherited Members

## 4.20.1 Detailed Description

Definition at line 17 of file MainWindow.hpp.

## 4.20.2 Constructor & Destructor Documentation

#### 4.20.2.1 MainWindow()

```
MainWindow::MainWindow (
            int width,
            int height )
```

Definition at line 8 of file MainWindow.cpp.

#### 4.20.2.2 ∼MainWindow()

```
MainWindow::~MainWindow ( )
```

Definition at line 26 of file MainWindow.cpp.

## 4.20.3 Member Function Documentation

#### 4.20.3.1 keyPressEvent()

```
void MainWindow::keyPressEvent (
            QKeyEvent * event )  [override], [protected]
```

Definition at line 111 of file MainWindow.cpp.

#### 4.20.3.2 paintEvent()

```
void MainWindow::paintEvent (
            QPaintEvent * event )  [override], [protected]
```

Definition at line 101 of file MainWindow.cpp.

#### 4.20.3.3 set_is_running()

```
void MainWindow::set_is_running (
            bool running )
```

Definition at line 79 of file MainWindow.cpp.

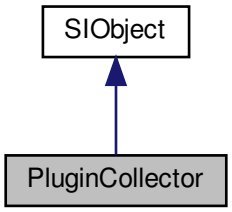
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/window/MainWindow.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/window/MainWindow.cpp

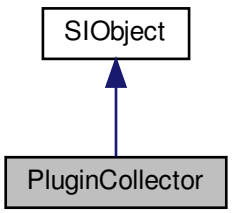
## 4.21 PluginCollector Class Reference

```
#include <PluginCollector.hpp>
```

Inheritance diagram for PluginCollector:




Collaboration diagram for PluginCollector:



### Public Member Functions

- PluginCollector ()
- ~PluginCollector ()=default
- void collect (const std::string &rel_path, std::vector< std::string > &files)

### Additional Inherited Members

## 4.21.1 Detailed Description

Definition at line 9 of file PluginCollector.hpp.

## 4.21.2 Constructor & Destructor Documentation

#### 4.21.2.1 PluginCollector()

```
PluginCollector::PluginCollector ( )  [inline]
```

Definition at line 12 of file PluginCollector.hpp.

#### 4.21.2.2 ~PluginCollector()

```
PluginCollector::~PluginCollector ( )  [default]
```

## 4.21.3 Member Function Documentation

#### 4.21.3.1 collect()

```
void PluginCollector::collect (
            const std::string & rel_path,
            std::vector< std::string > & files )
```

Definition at line 8 of file PluginCollector.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/PluginCollector.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/PluginCollector.cpp

## 4.22 Position Struct Reference

`#include <Vertex.hpp>`

### Public Attributes

- float x
- float y

### 4.22.1 Detailed Description

Definition at line 9 of file Vertex.hpp.

### 4.22.2 Member Data Documentation

#### 4.22.2.1 x

`float Position::x`

Definition at line 11 of file Vertex.hpp.

#### 4.22.2.2 y

`float Position::y`

Definition at line 12 of file Vertex.hpp.

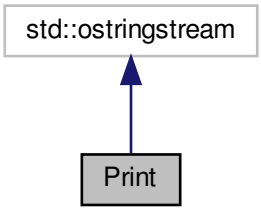
The documentation for this struct was generated from the following file:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Vertex.hpp

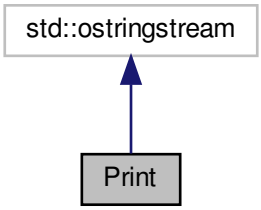
## 4.23 Print Class Reference

`#include <Print.hpp>`

Inheritance diagram for Print:




Collaboration diagram for Print:




### Public Member Functions

- Print ()=default
- ~Print ()

### Static Public Member Functions

- template<typename T >
  static void print (const std::vector< std::vector< T >> &v)
- template<typename T >
  static void print (const std::vector< T > &v)
- template<typename T1 , typename T2 >
  static void print (const std::map< T1, T2 > &map)
- template<typename T >
  static void print (const T &arg)

### 4.23.1 Detailed Description

Definition at line 14 of file Print.hpp.

### 4.23.2 Constructor & Destructor Documentation

#### 4.23.2.1 Print()

```
Print::Print ( )  [default]
```

#### 4.23.2.2 ~Print()

```
Print::~Print ( )  [inline]
```

Definition at line 19 of file Print.hpp.

### 4.23.3 Member Function Documentation

#### 4.23.3.1 print() [1/4]

```
template<typename T >
static void Print::print (
            const std::vector< std::vector< T >> & v )  [inline], [static]
```

Definition at line 26 of file Print.hpp.

#### 4.23.3.2 print() [2/4]

```
template<typename T >
static void Print::print (
            const std::vector< T > & v )  [inline], [static]
```

Definition at line 51 of file Print.hpp.

#### 4.23.3.3 print() [3/4]

```
template<typename T1 , typename T2 >
static void Print::print (
            const std::map< T1, T2 > & map )  [inline], [static]
```

Definition at line 71 of file Print.hpp.

#### 4.23.3.4 print() [4/4]

```
template<typename T >
static void Print::print (
            const T & arg )  [inline], [static]
```

Definition at line 82 of file Print.hpp.

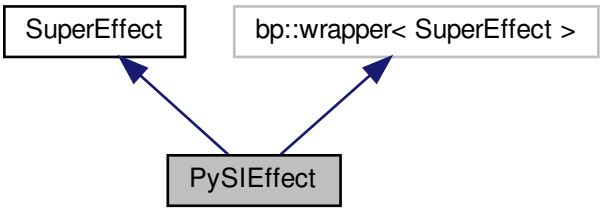
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/debug/Print.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/debug/Print.cpp

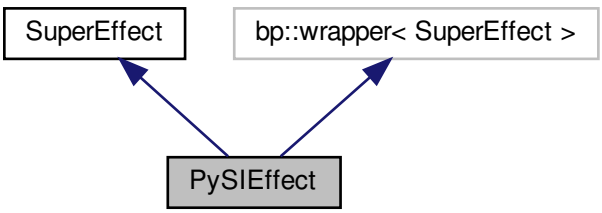
## 4.24 PySIEffect Class Reference

`#include <SuperEffect.hpp>`

Inheritance diagram for PySIEffect:




Collaboration diagram for PySIEffect:



### Public Member Functions

- int on_enter (bp::object &other) override
- int on_continuous (bp::object &other) override
- int on_leave (bp::object &other) override

## 4.24.1 Detailed Description

Definition at line 40 of file SuperEffect.hpp.

## 4.24.2 Member Function Documentation

### 4.24.2.1 on_continuous()

```
int PySIEffect::on_continuous (
            bp::object & other )  [override], [virtual]
```

Implements SuperEffect.

Definition at line 51 of file SuperEffect.cpp.

### 4.24.2.2 on_enter()

```
int PySIEffect::on_enter (
            bp::object & other )  [override], [virtual]
```

Implements SuperEffect.

Definition at line 46 of file SuperEffect.cpp.

### 4.24.2.3 on_leave()

```
int PySIEffect::on_leave (
            bp::object & other )  [override], [virtual]
```

Implements SuperEffect.

Definition at line 56 of file SuperEffect.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/pysi/SuperEffect.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/pysi/SuperEffect.cpp

## 4.25 PythonInvoker Class Reference

`#include <PythonInvoker.hpp>`

### Public Member Functions

- PythonInvoker ()
- ~PythonInvoker ()
- template<typename T >
  T invoke_extract_attribute (const bp::object &self, const std::string &attribute_name)
- template<typename T >
  T invoke_extract_attribute (const bp::list &self, int index)
- template<typename T >
  void invoke_set_attribute (bp::object &self, std::string &attribute_name, T &value, bool is_pointer=false)
- template<typename T >
  T invoke_function (bp::object &self, const std::string &function_name, bp::object &other)
- int invoke_collision_event_function (bp::object &self, bp::object &other, const std::string &function_name)
- int invoke_linking_event_function (bp::object &self, const std::string &capability, const bp::list &args)
- const bp::list retrieve_linking_event_args (const bp::object &self, const std::string &capability)
- void handle_python_error ()

### 4.25.1 Detailed Description

Definition at line 17 of file PythonInvoker.hpp.

### 4.25.2 Constructor & Destructor Documentation

#### 4.25.2.1 PythonInvoker()

`PythonInvoker::PythonInvoker ( ) [default]`

#### 4.25.2.2 ~PythonInvoker()

`PythonInvoker::~PythonInvoker ( ) [default]`

### 4.25.3 Member Function Documentation

#### 4.25.3.1 handle_python_error()

```
void PythonInvoker::handle_python_error ( )
```

Definition at line 68 of file PythonInvoker.cpp.

#### 4.25.3.2 invoke_collision_event_function()

```
int PythonInvoker::invoke_collision_event_function (
            bp::object & self,
            bp::object & other,
            const std::string & function_name )
```

Definition at line 10 of file PythonInvoker.cpp.

#### 4.25.3.3 invoke_extract_attribute() [1/2]

```
template<typename T >
T PythonInvoker::invoke_extract_attribute (
            const bp::object & self,
            const std::string & attribute_name )  [inline]
```

Definition at line 24 of file PythonInvoker.hpp.

#### 4.25.3.4 invoke_extract_attribute() [2/2]

```
template<typename T >
T PythonInvoker::invoke_extract_attribute (
            const bp::list & self,
            int index )  [inline]
```

Definition at line 39 of file PythonInvoker.hpp.

#### 4.25.3.5 invoke_function()

```
template<typename T >
T PythonInvoker::invoke_function (
            bp::object & self,
            const std::string & function_name,
            bp::object & other )  [inline]
```

Definition at line 70 of file PythonInvoker.hpp.

#### 4.25.3.6 invoke_linking_event_function()

```
int PythonInvoker::invoke_linking_event_function (
            bp::object & self,
            const std::string & capability,
            const bp::list & args )
```

Definition at line 43 of file PythonInvoker.cpp.

#### 4.25.3.7 invoke_set_attribute()

```
template<typename T >
void PythonInvoker::invoke_set_attribute (
            bp::object & self,
            std::string & attribute_name,
            T & value,
            bool is_pointer = false )  [inline]
```

Definition at line 54 of file PythonInvoker.hpp.

#### 4.25.3.8 retrieve_linking_event_args()

```
const bp::list PythonInvoker::retrieve_linking_event_args (
            const bp::object & self,
            const std::string & capability )
```

Definition at line 54 of file PythonInvoker.cpp.

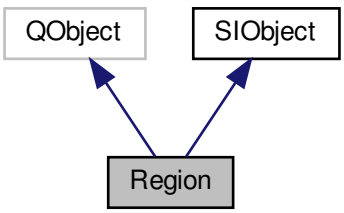
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/PythonInvoker.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/PythonInvoker.cpp

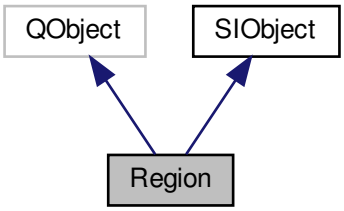
## 4.26 Region Class Reference

```
#include <Region.hpp>
```

Inheritance diagram for Region:



Collaboration diagram for Region:




## Public Member Functions

- Region (const std::vector< glm::vec3 > &contour, std::shared_ptr< bp::object > effect)
- ~Region ()
- const bool is_transformed () const
- void set_is_transformed (bool b)
- const std::string uuid () const
- bp::object & effect ()
- const std::unique_ptr< RegionMask > & mask () const
- const std::vector< glm::vec3 > & aabb ()
- const std::vector< glm::vec3 > & contour ()
- void set_aabb ()
- const std::string & texture_path () const
- void move (int x, int y)
- const glm::mat3x3 & transform () const
- int on_enter (bp::object &other)
- int on_continuous (bp::object &other)
- int on_leave (bp::object &other)
- Q_SIGNAL void LINK_SIGNAL (const std::string &uuid, const std::string &source_cap, const bp::list &py_list)
- Q_SLOT void LINK_SLOT (const std::string &uuid, const std::string &source_cap, const bp::list &py_list)
- void register_link_event (const std::string &uuid, const std::string &attribute)
- void register_link_event (const std::tuple< std::string, std::string > &link_event)
- bool is_link_event_registered (const std::string &uuid, const std::string &attribute)
- bool is_link_event_registered (const std::tuple< std::string, std::string > &link_event)
- void set_name (const std::string &name)
- const std::string & name () const

## Additional Inherited Members

### 4.26.1 Detailed Description

Definition at line 24 of file Region.hpp.

### 4.26.2 Constructor & Destructor Documentation

#### 4.26.2.1 Region()

```
Region::Region (
            const std::vector< glm::vec3 > & contour,
            std::shared_ptr< bp::object > effect )
```

Definition at line 14 of file Region.cpp.

#### 4.26.2.2 ~Region()

```
Region::~Region ( )
```

Definition at line 66 of file Region.cpp.

### 4.26.3 Member Function Documentation

#### 4.26.3.1 aabb()

```
const std::vector< glm::vec3 > & Region::aabb ( )
```

Definition at line 103 of file Region.cpp.

#### 4.26.3.2 contour()

```
const std::vector< glm::vec3 > & Region::contour ( )
```

Definition at line 108 of file Region.cpp.

#### 4.26.3.3 effect()

```
bp::object & Region::effect ( )
```

Definition at line 93 of file Region.cpp.

#### 4.26.3.4 is_link_event_registered() [1/2]

```
bool Region::is_link_event_registered (
            const std::string & uuid,
            const std::string & attribute )
```

Definition at line 234 of file Region.cpp.

#### 4.26.3.5 is_link_event_registered() [2/2]

```
bool Region::is_link_event_registered (
            const std::tuple< std::string, std::string > & link_event )
```

Definition at line 239 of file Region.cpp.

#### 4.26.3.6 is_transformed()

```
const bool Region::is_transformed ( ) const
```

Definition at line 78 of file Region.cpp.

#### 4.26.3.7 LINK_SIGNAL()

```
Q_SIGNAL void Region::LINK_SIGNAL (
            const std::string & uuid,
            const std::string & source_cap,
            const bp::list & py_list )
```

#### 4.26.3.8 LINK_SLOT()

```
void Region::LINK_SLOT (
            const std::string & uuid,
            const std::string & source_cap,
            const bp::list & py_list )
```

Definition at line 187 of file Region.cpp.

#### 4.26.3.9 mask()

```
const std::unique_ptr< RegionMask > & Region::mask ( ) const
```

Definition at line 98 of file Region.cpp.

#### 4.26.3.10 move()

```
void Region::move (
            int x,
            int y )
```

Definition at line 71 of file Region.cpp.

#### 4.26.3.11 name()

```
const std::string & Region::name ( ) const
```

Definition at line 249 of file Region.cpp.

#### 4.26.3.12 on_continuous()

```
int Region::on_continuous (
            bp::object & other )
```

Definition at line 162 of file Region.cpp.

#### 4.26.3.13 on_enter()

```
int Region::on_enter (
            bp::object & other )
```

Definition at line 149 of file Region.cpp.

#### 4.26.3.14 on_leave()

```
int Region::on_leave (
            bp::object & other )
```

Definition at line 175 of file Region.cpp.

#### 4.26.3.15 register_link_event() [1/2]

```
void Region::register_link_event (
            const std::string & uuid,
            const std::string & attribute )
```

Definition at line 224 of file Region.cpp.

#### 4.26.3.16 register_link_event() [2/2]

```
void Region::register_link_event (
            const std::tuple< std::string, std::string > & link_event )
```

Definition at line 229 of file Region.cpp.

#### 4.26.3.17 set_aabb()

```
void Region::set_aabb ( )
```

Definition at line 113 of file Region.cpp.

#### 4.26.3.18 set_is_transformed()

```
void Region::set_is_transformed (
            bool b )
```

Definition at line 83 of file Region.cpp.

#### 4.26.3.19 set_name()

```
void Region::set_name (
            const std::string & name )
```

Definition at line 244 of file Region.cpp.

#### 4.26.3.20 texture_path()

```
const std::string & Region::texture_path ( ) const
```

Definition at line 139 of file Region.cpp.

#### 4.26.3.21 transform()

```
const glm::mat3x3 & Region::transform ( ) const
```

Definition at line 144 of file Region.cpp.

#### 4.26.3.22 uuid()

```
const std::string Region::uuid ( ) const
```

Definition at line 88 of file Region.cpp.

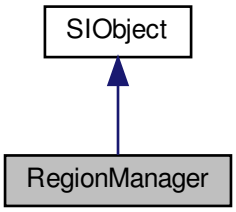
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/Region.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/Region.cpp

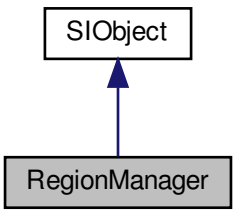
## 4.27 RegionManager Class Reference

```
#include <RegionManager.hpp>
```

Inheritance diagram for RegionManager:




Collaboration diagram for RegionManager:



### Public Member Functions

- ~RegionManager ()
- void add_region (const std::vector< glm::vec3 > &contour, std::shared_ptr< bp::object > effect, int region_uuid)
- std::vector< std::shared_ptr< Region > > & regions ()
- void update (const std::vector< int > &update_vertices)
- RegionManager ()

### Friends

- class SIGRunRegionManagerTest

### Additional Inherited Members

## 4.27.1 Detailed Description

Definition at line 15 of file RegionManager.hpp.

## 4.27.2 Constructor & Destructor Documentation

#### 4.27.2.1 ~RegionManager()

```
RegionManager::~RegionManager ( )
```

Definition at line 7 of file RegionManager.cpp.

#### 4.27.2.2 RegionManager()

```
RegionManager::RegionManager ( )
```

Definition at line 26 of file RegionManager.cpp.

## 4.27.3 Member Function Documentation

#### 4.27.3.1 add_region()

```
void RegionManager::add_region (
            const std::vector< glm::vec3 > & contour,
            std::shared_ptr< bp::object > effect,
            int region_uuid )
```

Definition at line 11 of file RegionManager.cpp.

#### 4.27.3.2 regions()

```
std::vector< std::shared_ptr< Region > > & RegionManager::regions ( )
```

Definition at line 16 of file RegionManager.cpp.

#### 4.27.3.3 update()

```
void RegionManager::update (
            const std::vector< int > & update_vertices )
```

Definition at line 21 of file RegionManager.cpp.

### 4.27.4 Friends And Related Function Documentation

#### 4.27.4.1 SIGRunRegionManagerTest

```
friend class SIGRunRegionManagerTest  [friend]
```

Definition at line 31 of file RegionManager.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/RegionManager.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/RegionManager.cpp

## 4.28 RegionMask Class Reference

RegionMask class which stores a bit array used for true collision testing.

```
#include <RegionMask.hpp>
```

### Public Member Functions

- RegionMask (int canvas_width, int canvas_height, const std::vector< glm::vec3 > &contour, const std::vector< glm::vec3 > &aabb)

  *constructor of the RegionMask class*
- RegionMask (const RegionMask &rm)

  *copy constructor*
- ~RegionMask ()

  *default destructor*
- int size () const

  *retrieve the size of the mask datastructure*
- void set_bit (int i)

  *set the bit at index i of d_values to one/true*
- void set_bit (const glm::vec3 &v)

  *set the bit at point v to one/true in d_values*
- void clear_bit (int i)

  *set the bit at index i of d_values to zero/false*
- void clear_bit (const glm::vec3 &v)

  *set the bit at point v of d_values to zero/false*
- int width () const
- int height () const
- void move (const glm::vec2 &v)

  *update the AABB relations according to desired translation of a parent Region*
- bool operator[ ] (int i) const

  *[] operator overloaded for returning the value of d_values at index i*
- bool operator[ ] (const glm::vec3 &v) const

  *[] operator overloaded for returning the value of d_values at point v*

### Friends

- class SIGRunRegionMaskTest

### 4.28.1 Detailed Description

RegionMask class which stores a bit array used for true collision testing.

Functionality
RegionMask class storing a bit array as std::vector<bool>. std::vector<bool> has a special implementation where its bool is stored in exactly one bit. See: https://en.cppreference.com/w/cpp/container/vector_bool This vector has the size of width * height of the AABB of the contour of the parent region. The array is filled with ones and zeroes according to a scanline algorithm. Every pixel which is part of the parent Region is set to one in that way. Others are left at 0. The array is relatively accessed according to the top left corner of that AABB.
Rationale:
The use of the AABB allows for creating a secondary coordinate system which is translated relatively to the parent coordinate system (canvas coordinate system). Therefore, each point which is to be tested with the mask is subtracted by the position vector of the AABB. In this way, that point in the canvas coordinate system is converted to the mask coordinate system. This leads to querying collosion occurrences relatively to the AABB. Through that, simple region translation does not require recomputation of the mask. Instead, the internal AABB is translated the same amount and the coordinate system conversion provides correct collision detection behaviour.

Definition at line 35 of file RegionMask.hpp.

### 4.28.2 Constructor & Destructor Documentation

#### 4.28.2.1 RegionMask() [1/2]

```
RegionMask::RegionMask (
            int canvas_width,
            int canvas_height,
            const std::vector< glm::vec3 > & contour,
            const std::vector< glm::vec3 > & aabb )
```

constructor of the RegionMask class

Constructor of the RegionMask class. Initializes all datastructures required to maintain a RegionMask for Collision↩ Detection according to parameters. Performs scanline algorithm for generation of the actual mask relatively to AABB of parent region.

**Parameters**

| | |
|---|---|
| *canvas_width* | int containing the width of the canvas |
| *canvas_height* | int containing the height of the canvas |
| *contour* | constant reference to a std::vector object containing glm::vec3 objects containing all the points of the contour of the parent Region |
| *aabb* | constant reference to a std::vector object containing glm::vec3 objects containing the four points of the AABB of the parent Region |

**See also**

d_canvas_width
d_canvas_height
d_tlc_aabb_x
d_tlc_aabb_y
d_brc_aabb_x
d_brc_aabb_y
d_width_aabb
d_height_aabb
d_values

Definition at line 28 of file RegionMask.cpp.

#### 4.28.2.2 RegionMask() [2/2]

```
RegionMask::RegionMask (
            const RegionMask & rm )
```

copy constructor

**Parameters**

| | |
|---|---|
| *rm* | the constant reference to a RegionMask object to be copied |

**See also**

d_canvas_width
d_canvas_height
d_tlc_aabb_x
d_tlc_aabb_y
d_brc_aabb_x
d_brc_aabb_y
d_width_aabb
d_height_aabb
d_values

Definition at line 68 of file RegionMask.cpp.

#### 4.28.2.3 ~RegionMask()

```
RegionMask::~RegionMask ( )
```

default destructor

Definition at line 86 of file RegionMask.cpp.

### 4.28.3 Member Function Documentation

#### 4.28.3.1 clear_bit() [1/2]

```
void RegionMask::clear_bit (
            int i )
```

set the bit at index i of d_values to zero/false

Sets the bit at index i of d_values to zero or false according to a bounds check.

**Parameters**

| | |
|---|---|
| *i* | int which contains the index of the bit to be set to zero/false in d_values |

**See also**

d_values

Definition at line 153 of file RegionMask.cpp.

#### 4.28.3.2 clear_bit() [2/2]

```
void RegionMask::clear_bit (
            const glm::vec3 & v )
```

set the bit at point v of d_values to zero/false

Sets the bit at point v of d_values to zero or false according to a bounds check. The bounds check is performed based on the actual index of the bit to set. The actual index is calculated according to AABB_WIDTH ∗ (v.y - A↩ ABB_TOP_LEFT_CORNER_Y) + v.x - AABB_TOP_LEFT_CORNER_X If the bounds check is negative, nothing happens.

**Parameters**

| | |
|---|---|
| *v* | a constant reference to a glm::vec3 object containing the corresponding coordinates of the point to a bit of d_values which is to be set to zero or false. |

**See also**

d_values
d_width_aabb
d_tlc_aabb_y
d_tlc_aabb_x

Definition at line 174 of file RegionMask.cpp.

#### 4.28.3.3 height()

```
int RegionMask::height ( ) const
```

**Returns**

the height of the AABB of the parent Region

**See also**

d_height_aabb

Definition at line 197 of file RegionMask.cpp.

#### 4.28.3.4 move()

```
void RegionMask::move (
            const glm::vec2 & v )
```

update the AABB relations according to desired translation of a parent Region

Use of RegionMasks occurs relatively to the AABB of its parent Region. Is the parent region moved / translated within the canvas, the AABB is also moved or translated. Therefore, the RegionMasks is upated according to that translation by storing the new translation parameters. In this way, the mask coordinate system is moved within the canvas coordinate system. So, after updating the AABB with the new translation values, the mask continues to function, due to its relative dependence on the AABB. Therefore, no recomputation is required.

**Parameters**

| | |
|---|---|
| *v* | a constant reference to a glm::vec2 object containg the translation vector |

Definition at line 259 of file RegionMask.cpp.

#### 4.28.3.5 operator[]() [1/2]

```
bool RegionMask::operator[] (
            int i ) const
```

[] operator overloaded for returning the value of d_values at index i

Overloads the [] operator. Retrieves the bool value at index i of d_values.

**Parameters**

| | |
|---|---|
| *i* | int containing the index |

**Returns**

a bool containing whether the queried bit is set or not in d_values

**See also**

d_values

Definition at line 213 of file RegionMask.cpp.

#### 4.28.3.6 operator[]() [2/2]

```
bool RegionMask::operator[] (
            const glm::vec3 & v ) const
```

[] operator overloaded for returning the value of d_values at point v

Overloads the [] operator. Retrieves the bool value at point v of d_values. The actual index is calculated according to AABB_WIDTH $*$ (v.y - AABB_TOP_LEFT_CORNER_Y) + v.x - AABB_TOP_LEFT_CORNER_X.

**Parameters**

| | |
|---|---|
| *v* | a constant reference to a glm::vec3 object containing the corresponding coordinates of the point to a bit of d_values which is to be tested whether the queried bit is set or not in d_values. |

**Returns**

a bool containing whether the queried bit is set or not in d_values

**See also**

d_values
d_width_aabb
d_tlc_aabb_x
d_tlc_aabb_y

Definition at line 237 of file RegionMask.cpp.

#### 4.28.3.7 set_bit() [1/2]

```
void RegionMask::set_bit (
            int i )
```

set the bit at index i of d_values to one/true

Sets the bit at index i of d_values to one or true according to a bounds check.

**Parameters**

| | |
|---|---|
| *i* | int which contains the index of the bit to be set to one/true in d_values |

**See also**

d_values

Definition at line 115 of file RegionMask.cpp.

#### 4.28.3.8 set_bit() [2/2]

```
void RegionMask::set_bit (
            const glm::vec3 & v )
```

set the bit at point v to one/true in d_values

Sets the bit at point v of d_values to one or true according to a bounds check. The bounds check is performed based on the actual index of the bit to be set. The actual index is calculated according to AABB_WIDTH $*$ (v.y - AABB_TOP_LEFT_CORNER_Y) + v.x - AABB_TOP_LEFT_CORNER_X If the bounds check is negative, the bit will be set to false.

**Parameters**

| | |
|---|---|
| *v* | a constant reference to a glm::vec3 object containing the corresponding coordinates of the point to a bit of d_values which is to be set to one or true. |

**See also**

d_values
d_width_aabb
d_tlc_aabb_y
d_tlc_aabb_x

Definition at line 136 of file RegionMask.cpp.

#### 4.28.3.9 size()

```
int RegionMask::size ( ) const
```

retrieve the size of the mask datastructure

Retrieves the size of the mask datastructure. This datastructure is called d_values.

**Returns**

the size of d_values

**See also**

d_values

Definition at line 101 of file RegionMask.cpp.

#### 4.28.3.10 width()

```
int RegionMask::width ( ) const
```

**Returns**

the width of the AABB of the parent Region

**See also**

d_width_aabb

Definition at line 187 of file RegionMask.cpp.

### 4.28.4 Friends And Related Function Documentation

#### 4.28.4.1 SIGRunRegionMaskTest

```
friend class SIGRunRegionMaskTest  [friend]
```

Definition at line 106 of file RegionMask.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionMask.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionMask.cpp

## 4.29 RegionRepresentation Struct Reference

```
#include <RegionRepresentation.hpp>
```

### Public Member Functions

- RegionRepresentation (const std::vector< glm::vec3 > &c, const glm::mat3x3 &t, const QColor &col, const std::string &tex_path)
- RegionRepresentation (const std::vector< glm::vec3 > &c, const glm::mat3x3 &t, int r_, int g_, int b_, int a_, const std::string &tex_path)
- void update (const glm::mat3x3 &transform)
- RegionRepresentation (std::vector< TesselationPatch * > ps, const glm::mat3x3 &t, const std::string &tex↩ _path, int cs, glm::vec4 dest_rect, const glm::vec4 _uv)

### Public Attributes

- int r
- int g
- int b
- int a
- QColor color
- std::string texture_path
- QPolygonF poly
- QPainterPath fill
- std::vector< TesselationPatch * > patches
- glm::mat3x3 transform
- int contour_size
- glm::vec4 destination_rect
- glm::vec4 uv

### 4.29.1 Detailed Description

Definition at line 9 of file RegionRepresentation.hpp.

### 4.29.2 Constructor & Destructor Documentation

#### 4.29.2.1 RegionRepresentation() [1/3]

```
RegionRepresentation::RegionRepresentation (
            const std::vector< glm::vec3 > & c,
            const glm::mat3x3 & t,
            const QColor & col,
            const std::string & tex_path )  [inline]
```

Definition at line 11 of file RegionRepresentation.hpp.

#### 4.29.2.2 RegionRepresentation() [2/3]

```
RegionRepresentation::RegionRepresentation (
            const std::vector< glm::vec3 > & c,
            const glm::mat3x3 & t,
            int r_,
            int g_,
            int b_,
            int a_,
            const std::string & tex_path )  [inline]
```

Definition at line 30 of file RegionRepresentation.hpp.

#### 4.29.2.3 RegionRepresentation() [3/3]

```
RegionRepresentation::RegionRepresentation (
            std::vector< TesselationPatch * > ps,
            const glm::mat3x3 & t,
            const std::string & tex_path,
            int cs,
            glm::vec4 dest_rect,
            const glm::vec4 _uv )  [inline]
```

Definition at line 14 of file RegionRepresentation.hpp.

### 4.29.3 Member Function Documentation

#### 4.29.3.1 update()

```
void RegionRepresentation::update (
            const glm::mat3x3 & transform )  [inline]
```

Definition at line 49 of file RegionRepresentation.hpp.

### 4.29.4 Member Data Documentation

#### 4.29.4.1 a

```
int RegionRepresentation::a
```

Definition at line 69 of file RegionRepresentation.hpp.

#### 4.29.4.2 b

```
int RegionRepresentation::b
```

Definition at line 68 of file RegionRepresentation.hpp.

#### 4.29.4.3 color

```
QColor RegionRepresentation::color
```

Definition at line 70 of file RegionRepresentation.hpp.

#### 4.29.4.4 contour_size

```
int RegionRepresentation::contour_size
```

Definition at line 26 of file RegionRepresentation.hpp.

#### 4.29.4.5 destination_rect

```
glm::vec4 RegionRepresentation::destination_rect
```

Definition at line 27 of file RegionRepresentation.hpp.

#### 4.29.4.6 fill

```
QPainterPath RegionRepresentation::fill
```

Definition at line 73 of file RegionRepresentation.hpp.

#### 4.29.4.7 g

```
int RegionRepresentation::g
```

Definition at line 67 of file RegionRepresentation.hpp.

#### 4.29.4.8 patches

```
std::vector<TesselationPatch*> RegionRepresentation::patches
```

Definition at line 23 of file RegionRepresentation.hpp.

#### 4.29.4.9 poly

```
QPolygonF RegionRepresentation::poly
```

Definition at line 72 of file RegionRepresentation.hpp.

#### 4.29.4.10 r

```
int RegionRepresentation::r
```

Definition at line 66 of file RegionRepresentation.hpp.

#### 4.29.4.11 texture_path

```
std::string RegionRepresentation::texture_path
```

Definition at line 71 of file RegionRepresentation.hpp.

#### 4.29.4.12 transform

```
glm::mat3x3 RegionRepresentation::transform
```

Definition at line 24 of file RegionRepresentation.hpp.

#### 4.29.4.13 uv

```
glm::vec4 RegionRepresentation::uv
```

Definition at line 28 of file RegionRepresentation.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/region/RegionRepresentation.hpp

## 4.30 RegionResampler Class Reference

```
#include <RegionResampler.hpp>
```

### Static Public Member Functions

- static void resample (std::vector< glm::vec3 > &out, const std::vector< glm::vec3 > &in, int step_count=64)

### Friends

- class SIGRunRegionResamplerTest

### 4.30.1 Detailed Description

Definition at line 9 of file RegionResampler.hpp.

### 4.30.2 Member Function Documentation

#### 4.30.2.1 resample()

```
void RegionResampler::resample (
            std::vector< glm::vec3 > & out,
            const std::vector< glm::vec3 > & in,
            int step_count = 64 )  [static]
```

Definition at line 5 of file RegionResampler.cpp.

### 4.30.3 Friends And Related Function Documentation

#### 4.30.3.1 SIGRunRegionResamplerTest

```
friend class SIGRunRegionResamplerTest  [friend]
```

Definition at line 18 of file RegionResampler.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionResampler.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionResampler.cpp

## 4.31 RegionTransform Class Reference

RegionTransform class storing the relative translation, rotation and scale of a contour.

```
#include <RegionTransform.hpp>
```

### Public Member Functions

- RegionTransform ()
  *default constructor initializing instance variables to default values*
- ~RegionTransform ()
  *default destructor*
- void update (const glm::vec2 &translation=glm::vec2(0, 0), float angle=0, float scale=1)
  *central function to update transformation matrix with new, relative translation, relative rotation and absolute scale values*
- const glm::mat3x3 & transform ()
- const glm::vec3 & operator[ ] (int index)
  *overloading of [] operator*

### 4.31.1 Detailed Description

RegionTransform class storing the relative translation, rotation and scale of a contour.

This class stores the relative translation, rotation and scale of a contour. The initial contour remains unchanged and change in one of those three aspects does mutate this transform but not the initial contour. The translation, rotation and scale are stored as a 3x3 transformation matrix. The transformation matrix is stored ROW MAJOR and requires LEFT pr PRE-Multiplication. Therefore, multiplications with points look such as : $p * T$, where p is a point and T is the transformation matrix. Due to matrix multiplications being not commutative, $T * p$ will not yield desired results.

**See also**

d_translation
d_rotation
d_scale
d_transform
d_angle

Definition at line 34 of file RegionTransform.hpp.

### 4.31.2 Constructor & Destructor Documentation

#### 4.31.2.1 RegionTransform()

```
RegionTransform::RegionTransform ( )
```

default constructor initializing instance variables to default values

Default constructor. Initializes all matrix objects to identity matrices. Sets cumulative angle to 0

**See also**

d_transform
d_translation
d_rotation
d_scale \scale d_angle

Definition at line 17 of file RegionTransform.cpp.

#### 4.31.2.2 ~RegionTransform()

```
RegionTransform::~RegionTransform ( )
```

default destructor

Default destructor.

Definition at line 30 of file RegionTransform.cpp.

### 4.31.3 Member Function Documentation

#### 4.31.3.1 operator[]()

```
const glm::vec3 & RegionTransform::operator[] (
            int index )
```

overloading of [] operator

Overloading of [] operator. Makes it easier to use the transformation matrix stored in this class. This function returns a constant glm::vec3 reference which itself is subscriptable with the [] operator.

**Parameters**

| | |
|---|---|
| *index* | an integer containing the index of the row of the transformation matrix to be retrieved. |

**Returns**

a constant reference of glm::vec3 object containing the queried row of the transformation matrix

**See also**

d_transform

Definition at line 96 of file RegionTransform.cpp.

#### 4.31.3.2 transform()

```
const glm::mat3x3 & RegionTransform::transform ( )
```

**Returns**

a constant reference to a glm::mat3x3 object containing the current transformation matrix

**See also**

d_transform

Definition at line 78 of file RegionTransform.cpp.

#### 4.31.3.3 update()

```
void RegionTransform::update (
            const glm::vec2 & translation = glm::vec2(0, 0),
            float angle = 0,
            float scale = 1 )
```

central function to update transformation matrix with new, relative translation, relative rotation and absolute scale values

Updates translation matrix T, rotation matrix R, and scale matrix S according to the given parameters. Too small angle increments are ignored to save computations of required trigonometric functions. Computes a the new transformation matrix according to $T * R * S$.

**Parameters**

| | |
|---|---|
| *translation* | a constant reference to a glm::vec2 datastructure containing the new, relative translation of the parent contour |
| *angle* | a float containing the new relative angle of the parent contour according to x-axis |
| *scale* | a float containing the new absolute scale factor of the contour |

**See also**

d_translation
d_angle
d_rotation
d_scale
d_transform

Definition at line 50 of file RegionTransform.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionTransform.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionTransform.cpp

## 4.32 RenderBatch Class Reference

`#include <SpriteBatch.hpp>`

### Public Member Functions

- RenderBatch (GLuint off, GLuint vertices, GLuint off2, GLuint vertices2, GLuint tex)

### Public Attributes

- GLuint offset
- GLuint num_vertices
- GLuint offset2
- GLuint num_vertices2
- GLuint texture

### 4.32.1 Detailed Description

Definition at line 12 of file SpriteBatch.hpp.

### 4.32.2 Constructor & Destructor Documentation

#### 4.32.2.1 RenderBatch()

```
RenderBatch::RenderBatch (
            GLuint off,
            GLuint vertices,
            GLuint off2,
            GLuint vertices2,
            GLuint tex )  [inline]
```

Definition at line 21 of file SpriteBatch.hpp.

### 4.32.3 Member Data Documentation

#### 4.32.3.1 num_vertices

```
GLuint RenderBatch::num_vertices
```

Definition at line 16 of file SpriteBatch.hpp.

#### 4.32.3.2 num_vertices2

```
GLuint RenderBatch::num_vertices2
```

Definition at line 18 of file SpriteBatch.hpp.

#### 4.32.3.3 offset

```
GLuint RenderBatch::offset
```

Definition at line 15 of file SpriteBatch.hpp.

#### 4.32.3.4 offset2

```
GLuint RenderBatch::offset2
```

Definition at line 17 of file SpriteBatch.hpp.

#### 4.32.3.5 texture

```
GLuint RenderBatch::texture
```

Definition at line 19 of file SpriteBatch.hpp.

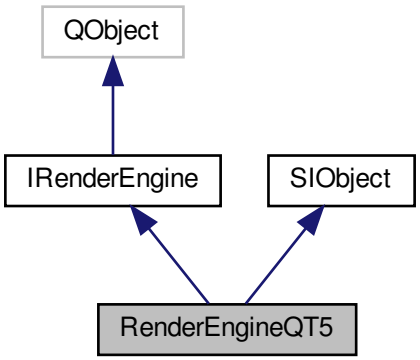
The documentation for this class was generated from the following file:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/SpriteBatch.hpp

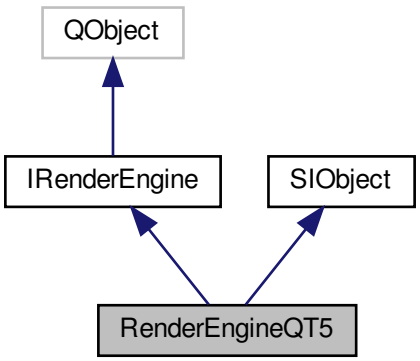
## 4.33 RenderEngineQT5 Class Reference

```
#include <RenderEngineQt5.hpp>
```

Inheritance diagram for RenderEngineQT5:




Collaboration diagram for RenderEngineQT5:



### Public Member Functions

- RenderEngineQT5 ()
- ~RenderEngineQT5 ()
- void start (int width, int height) override
- void run () override
- void pause () override

### Additional Inherited Members

## 4.33.1 Detailed Description

Definition at line 12 of file RenderEngineQt5.hpp.

## 4.33.2 Constructor & Destructor Documentation

### 4.33.2.1 RenderEngineQT5()

```
RenderEngineQT5::RenderEngineQT5 ( )
```

Definition at line 5 of file RenderEngineQt5.cpp.

### 4.33.2.2 ~RenderEngineQT5()

```
RenderEngineQT5::~RenderEngineQT5 ( )
```

Definition at line 10 of file RenderEngineQt5.cpp.

## 4.33.3 Member Function Documentation

### 4.33.3.1 pause()

```
void RenderEngineQT5::pause ( )  [override], [virtual]
```

Implements IRenderEngine.

Definition at line 28 of file RenderEngineQt5.cpp.

#### 4.33.3.2 run()

```
void RenderEngineQT5::run ( )  [override], [virtual]
```

Implements IRenderEngine.

Definition at line 23 of file RenderEngineQt5.cpp.

#### 4.33.3.3 start()

```
void RenderEngineQT5::start (
            int width,
            int height )  [override], [virtual]
```

Implements IRenderEngine.

Definition at line 17 of file RenderEngineQt5.cpp.

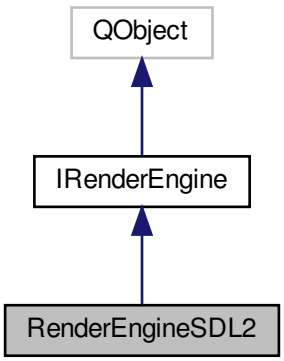
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/RenderEngineQt5.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/RenderEngineQt5.cpp

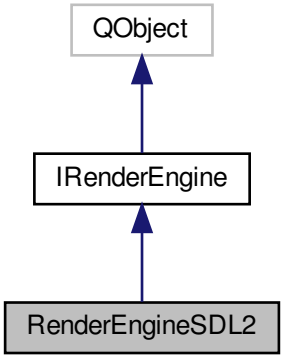
## 4.34 RenderEngineSDL2 Class Reference

```
#include <RenderEngineSdl2.hpp>
```

Inheritance diagram for RenderEngineSDL2:



Collaboration diagram for RenderEngineSDL2:




### Public Member Functions

- RenderEngineSDL2 ()
- ~RenderEngineSDL2 ()
- void start (int width, int height, int argc, char **argv) override
- void run () override
- void pause () override

## 4.34.1 Detailed Description

Definition at line 28 of file RenderEngineSdl2.hpp.

## 4.34.2 Constructor & Destructor Documentation

### 4.34.2.1 RenderEngineSDL2()

```
RenderEngineSDL2::RenderEngineSDL2 ( )
```

Definition at line 10 of file RenderEngineSdl2.cpp.

### 4.34.2.2 ~RenderEngineSDL2()

```
RenderEngineSDL2::~RenderEngineSDL2 ( )
```

Definition at line 20 of file RenderEngineSdl2.cpp.

### 4.34.3 Member Function Documentation

#### 4.34.3.1 pause()

```
void RenderEngineSDL2::pause ( )  [override], [virtual]
```

Implements IRenderEngine.

Definition at line 122 of file RenderEngineSdl2.cpp.

#### 4.34.3.2 run()

```
void RenderEngineSDL2::run ( )  [override], [virtual]
```

Implements IRenderEngine.

Definition at line 63 of file RenderEngineSdl2.cpp.

#### 4.34.3.3 start()

```
void RenderEngineSDL2::start (
            int width,
            int height,
            int argc,
            char ** argv )  [override]
```

Definition at line 40 of file RenderEngineSdl2.cpp.

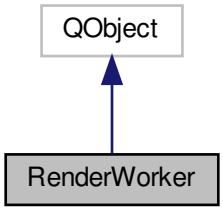
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/RenderEngineSdl2.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/RenderEngineSdl2.cpp

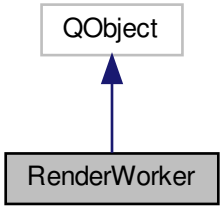
## 4.35 RenderWorker Class Reference

`#include <RenderWorker.hpp>`

Inheritance diagram for RenderWorker:




Collaboration diagram for RenderWorker:




### Public Member Functions

- RenderWorker (int width, int height, IRenderEngine *ire)
- Q_SLOT void render ()
- Q_SIGNAL void finished ()

### 4.35.1 Detailed Description

Definition at line 9 of file RenderWorker.hpp.

### 4.35.2 Constructor & Destructor Documentation

#### 4.35.2.1 RenderWorker()

```
RenderWorker::RenderWorker (
            int width,
            int height,
            IRenderEngine * ire )
```

Definition at line 5 of file RenderWorker.cpp.

### 4.35.3 Member Function Documentation

#### 4.35.3.1 finished()

```
Q_SIGNAL void RenderWorker::finished ( )
```

#### 4.35.3.2 render()

```
void RenderWorker::render ( )
```

Definition at line 14 of file RenderWorker.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/concurrency/RenderWorker.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/concurrency/RenderWorker.cpp

## 4.36 ResourceManager Class Reference

```
#include <ResourceManager.hpp>
```

### Static Public Member Functions

- static GLTexture texture (const std::string &texture_path)

### 4.36.1 Detailed Description

Definition at line 10 of file ResourceManager.hpp.

### 4.36.2 Member Function Documentation

#### 4.36.2.1 texture()

```
GLTexture ResourceManager::texture (
            const std::string & texture_path )  [static]
```

Definition at line 7 of file ResourceManager.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/ResourceManager.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/ResourceManager.cpp

## 4.37 RingBuffer< T > Class Template Reference

`#include <RingBuffer.hpp>`

### Public Member Functions

- RingBuffer (int size)
- ~RingBuffer ()=default
- void push_back (const T &data)
- const T & get ()
- bool find (const T &data) const
- void clear ()
- bool empty () const
- int size () const
- int max_size () const
- bool operator & (const T &value) const
- void operator<< (const T &value)

### 4.37.1 Detailed Description

**template<typename T>**
**class RingBuffer< T >**

Definition at line 10 of file RingBuffer.hpp.

### 4.37.2 Constructor & Destructor Documentation

#### 4.37.2.1 RingBuffer()

```
template<typename T>
RingBuffer< T >::RingBuffer (
            int size )  [inline], [explicit]
```

Definition at line 13 of file RingBuffer.hpp.

#### 4.37.2.2 ~RingBuffer()

```
template<typename T>
RingBuffer< T >::~RingBuffer ( )  [default]
```

### 4.37.3 Member Function Documentation

#### 4.37.3.1 clear()

```
template<typename T>
void RingBuffer< T >::clear ( )  [inline]
```

Definition at line 50 of file RingBuffer.hpp.

#### 4.37.3.2 empty()

```
template<typename T>
bool RingBuffer< T >::empty ( ) const  [inline]
```

Definition at line 56 of file RingBuffer.hpp.

#### 4.37.3.3 find()

```
template<typename T>
bool RingBuffer< T >::find (
            const T & data ) const  [inline]
```

Definition at line 45 of file RingBuffer.hpp.

#### 4.37.3.4 get()

```
template<typename T>
const T& RingBuffer< T >::get ( )  [inline]
```

Definition at line 35 of file RingBuffer.hpp.

#### 4.37.3.5 max_size()

```
template<typename T>
int RingBuffer< T >::max_size ( ) const  [inline]
```

Definition at line 66 of file RingBuffer.hpp.

#### 4.37.3.6 operator &()

```
template<typename T>
bool RingBuffer< T >::operator & (
            const T & value ) const  [inline]
```

Definition at line 71 of file RingBuffer.hpp.

#### 4.37.3.7 operator<<()

```
template<typename T>
void RingBuffer< T >::operator<< (
            const T & value )  [inline]
```

Definition at line 76 of file RingBuffer.hpp.

#### 4.37.3.8 push_back()

```
template<typename T>
void RingBuffer< T >::push_back (
            const T & data )  [inline]
```

Definition at line 25 of file RingBuffer.hpp.

#### 4.37.3.9 size()

```
template<typename T>
int RingBuffer< T >::size ( ) const  [inline]
```

Definition at line 61 of file RingBuffer.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/util/RingBuffer.hpp

## 4.38 Scripting Class Reference

`#include <Scripting.hpp>`

### Public Member Functions

- Scripting ()
- ~Scripting ()
- bp::object si_plugin (std::string &module_name, std::string &path, std::string &class_name)
- std::string load_plugin_source (const char *source)
- void load_class_names (std::vector< std::string > &classes, const std::string &path)
- bp::object import (const std::string &module, const std::string &path)

### Friends

- std::ostream & operator<< (std::ostream &os, const Scripting &scripting)

### 4.38.1 Detailed Description

Definition at line 13 of file Scripting.hpp.

### 4.38.2 Constructor & Destructor Documentation

#### 4.38.2.1 Scripting()

`Scripting::Scripting ( )`

Definition at line 11 of file Scripting.cpp.

#### 4.38.2.2 ~Scripting()

`Scripting::~Scripting ( )`

Definition at line 21 of file Scripting.cpp.

### 4.38.3 Member Function Documentation

#### 4.38.3.1 import()

```
bp::object Scripting::import (
            const std::string & module,
            const std::string & path )
```

Definition at line 95 of file Scripting.cpp.

#### 4.38.3.2 load_class_names()

```
void Scripting::load_class_names (
            std::vector< std::string > & classes,
            const std::string & path )
```

Definition at line 65 of file Scripting.cpp.

#### 4.38.3.3 load_plugin_source()

```
std::string Scripting::load_plugin_source (
            const char * source )
```

Definition at line 29 of file Scripting.cpp.

#### 4.38.3.4 si_plugin()

```
bp::object Scripting::si_plugin (
            std::string & module_name,
            std::string & path,
            std::string & class_name )
```

Definition at line 24 of file Scripting.cpp.

### 4.38.4 Friends And Related Function Documentation

#### 4.38.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const Scripting & scripting )  [friend]
```

Definition at line 110 of file Scripting.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/Scripting.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/Scripting.cpp

## 4.39 SIGRun Class Reference

SIGRun class serving as entry point of an SI environment.

`#include <SIGRun.hpp>`

### Public Member Functions

- SIGRun ()
  *constructor*
- ~SIGRun ()
  *destructor*
- int exec (int argc, char **argv, IRenderEngine *ire)
  *entry point of SIGRun*

### Static Public Member Functions

- static int quit ()
  *exit SIGRun*

### 4.39.1 Detailed Description

SIGRun class serving as entry point of an SI environment.

This class serves as the entry point of an SI environment. It is directly exposed in SI.hpp. An instance of this class is used to launch an SI environment.

**See also**

up_core

Definition at line 18 of file SIGRun.hpp.

### 4.39.2 Constructor & Destructor Documentation

#### 4.39.2.1 SIGRun()

`SIGRun::SIGRun ( )`

constructor

Constructor of SIGRun class. Used for instantiating objects.

Definition at line 19 of file SIGRun.cpp.

#### 4.39.2.2 ~SIGRun()

```
SIGRun::~SIGRun ( )
```

destructor

Destructor of SIGRun class. Used for destroying objects.

Definition at line 30 of file SIGRun.cpp.

### 4.39.3 Member Function Documentation

#### 4.39.3.1 exec()

```
int SIGRun::exec (
            int argc,
            char ** argv,
            IRenderEngine * ire )
```

entry point of SIGRun

Entry point of SIGRun initializing all further systems.

**Parameters**

| | |
|---|---|
| *argc* | cli argc |
| *argv* | cli argv |

Definition at line 42 of file SIGRun.cpp.

#### 4.39.3.2 quit()

```
int SIGRun::quit ( )  [static]
```

exit SIGRun

static exit function of SIGRun terminating all other systems

Definition at line 54 of file SIGRun.cpp.

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

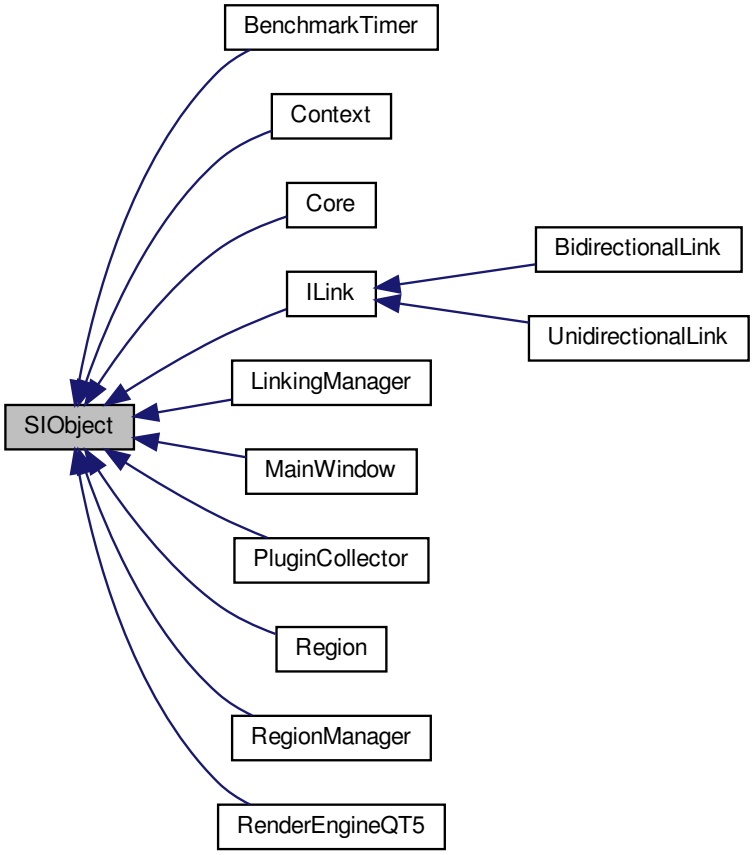
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/SIGRun.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/SIGRun.cpp

## 4.40 SIObject Class Reference

A meta class from which other classes are derived from to register them as SIObject meta types.

`#include <SIObject.hpp>`

Inheritance diagram for SIObject:




### Public Member Functions

- SIObject ()=default

  *default constructor*
- ~SIObject ()=default

  *default destructor*
- const std::string & meta_type () const

  *function for retrieving meta type name*
- const std::string & origin () const

### Protected Attributes

- std::string d_meta_type

  *a std::string containing the name of the class to be registered as SIObject meta type*
- std::string d_origin

## 4.40.1 Detailed Description

A meta class from which other classes are derived from to register them as SIObject meta types.

This class enables registering other classes as SIObject meta types. This is currently achieved by storing std::strings containing the classes individual names. Currently, this meta typing is only used for Logging.

**See also**

Log::Log
d_meta_type

Definition at line 55 of file SIObject.hpp.

## 4.40.2 Constructor & Destructor Documentation

### 4.40.2.1 SIObject()

```
SIObject::SIObject ( )  [default]
```

default constructor

### 4.40.2.2 ~SIObject()

```
SIObject::~SIObject ( )  [default]
```

default destructor

## 4.40.3 Member Function Documentation

#### 4.40.3.1 meta_type()

```
const std::string& SIObject::meta_type ( ) const  [inline]
```

function for retrieving meta type name

The function for retrieving meta type name in a constant manner. Therefore, the instance calling this function will not mutate.

**Returns**

d_meta_type a const std::string reference of the type name of the clas

Definition at line 75 of file SIObject.hpp.

#### 4.40.3.2 origin()

```
const std::string& SIObject::origin ( ) const  [inline]
```

Definition at line 80 of file SIObject.hpp.

### 4.40.4 Member Data Documentation

#### 4.40.4.1 d_meta_type

```
std::string SIObject::d_meta_type  [protected]
```

a std::string containing the name of the class to be registered as SIObject meta type

Definition at line 89 of file SIObject.hpp.

#### 4.40.4.2 d_origin

```
std::string SIObject::d_origin  [protected]
```

Definition at line 90 of file SIObject.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/SIObject.hpp

## 4.41 SpriteBatch Class Reference

`#include <SpriteBatch.hpp>`

### Public Member Functions

- SpriteBatch ()
- ~SpriteBatch ()
- void initialize ()
- void render (const std::map< std::string, RegionRepresentation * > &regions, const std::vector< glm::vec2 > &partial_contour, const GLfloat *camera_matrix)
- void set_draw_mode (GLenum mode)

### 4.41.1 Detailed Description

Definition at line 26 of file SpriteBatch.hpp.

### 4.41.2 Constructor & Destructor Documentation

#### 4.41.2.1 SpriteBatch()

`SpriteBatch::SpriteBatch ( )`

Definition at line 9 of file SpriteBatch.cpp.

#### 4.41.2.2 ~SpriteBatch()

`SpriteBatch::~SpriteBatch ( )`

Definition at line 17 of file SpriteBatch.cpp.

### 4.41.3 Member Function Documentation

#### 4.41.3.1 initialize()

`void SpriteBatch::initialize ( )`

Definition at line 26 of file SpriteBatch.cpp.

#### 4.41.3.2 render()

```
void SpriteBatch::render (
            const std::map< std::string, RegionRepresentation * > & regions,
            const std::vector< glm::vec2 > & partial_contour,
            const GLfloat * camera_matrix )
```

Definition at line 81 of file SpriteBatch.cpp.

#### 4.41.3.3 set_draw_mode()

```
void SpriteBatch::set_draw_mode (
            GLenum mode )
```

Definition at line 199 of file SpriteBatch.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/SpriteBatch.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/SpriteBatch.cpp

## 4.42 StoppableTask Class Reference

`#include <StoppableTask.hpp>`

### Public Member Functions

- StoppableTask ()
- StoppableTask (StoppableTask &&obj)
- StoppableTask & operator= (StoppableTask &&obj)
- virtual void run ()=0
- void operator() ()
- void set_is_running (bool is_running)
- bool is_stop_requested ()
- void stop ()

### Protected Attributes

- bool d_is_running = false

### 4.42.1 Detailed Description

Definition at line 8 of file StoppableTask.hpp.

### 4.42.2 Constructor & Destructor Documentation

#### 4.42.2.1 StoppableTask() [1/2]

```
StoppableTask::StoppableTask ( )
```

Definition at line 5 of file StoppableTask.cpp.

#### 4.42.2.2 StoppableTask() [2/2]

```
StoppableTask::StoppableTask (
            StoppableTask && obj )  [explicit]
```

Definition at line 10 of file StoppableTask.cpp.

### 4.42.3 Member Function Documentation

#### 4.42.3.1 is_stop_requested()

```
bool StoppableTask::is_stop_requested ( )
```

Definition at line 34 of file StoppableTask.cpp.

#### 4.42.3.2 operator()()

```
void StoppableTask::operator() ( )
```

Definition at line 24 of file StoppableTask.cpp.

#### 4.42.3.3 operator=()

```
StoppableTask & StoppableTask::operator= (
            StoppableTask && obj )
```

Definition at line 16 of file StoppableTask.cpp.

#### 4.42.3.4 run()

```
virtual void StoppableTask::run ( )  [pure virtual]
```

#### 4.42.3.5 set_is_running()

```
void StoppableTask::set_is_running (
            bool is_running )
```

Definition at line 29 of file StoppableTask.cpp.

#### 4.42.3.6 stop()

```
void StoppableTask::stop ( )
```

Definition at line 40 of file StoppableTask.cpp.

### 4.42.4 Member Data Documentation

#### 4.42.4.1 d_is_running

```
bool StoppableTask::d_is_running = false  [protected]
```

Definition at line 29 of file StoppableTask.hpp.

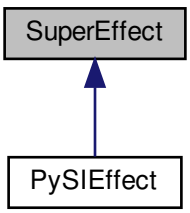
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/concurrency/StoppableTask.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/concurrency/StoppableTask.cpp

## 4.43 SuperEffect Class Reference

```
#include <SuperEffect.hpp>
```

Inheritance diagram for SuperEffect:



### Public Member Functions

- virtual int on_enter (bp::object &other)=0
- virtual int on_continuous (bp::object &other)=0
- virtual int on_leave (bp::object &other)=0

### 4.43.1 Detailed Description

Definition at line 32 of file SuperEffect.hpp.

### 4.43.2 Member Function Documentation

#### 4.43.2.1 on_continuous()

```
virtual int SuperEffect::on_continuous (
            bp::object & other )  [pure virtual]
```

Implemented in PySIEffect.

#### 4.43.2.2 on_enter()

```
virtual int SuperEffect::on_enter (
            bp::object & other )  [pure virtual]
```

Implemented in PySIEffect.

#### 4.43.2.3 on_leave()

```
virtual int SuperEffect::on_leave (
            bp::object & other )  [pure virtual]
```

Implemented in PySIEffect.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/pysi/SuperEffect.hpp

## 4.44 TesselationPatch Class Reference

```
#include <TessellationPatch.hpp>
```

### Public Member Functions

- TesselationPatch (const glm::vec3 &A, const glm::vec3 &B, const glm::vec3 &C)
- ~TesselationPatch ()
- void set_abc (const glm::vec3 &A, const glm::vec3 &B, const glm::vec3 &C)
- void move (const glm::vec3 &delta)
- const glm::vec3 & a () const
- const glm::vec3 & b () const
- const glm::vec3 & c () const
- std::vector< glm::vec3 > vertices ()

## 4.44.1 Detailed Description

Definition at line 10 of file TessellationPatch.hpp.

## 4.44.2 Constructor & Destructor Documentation

### 4.44.2.1 TesselationPatch()

```
TesselationPatch::TesselationPatch (
            const glm::vec3 & A,
            const glm::vec3 & B,
            const glm::vec3 & C )
```

Definition at line 5 of file TessellationPatch.cpp.

### 4.44.2.2 ~TesselationPatch()

```
TesselationPatch::~TesselationPatch ( )
```

Definition at line 11 of file TessellationPatch.cpp.

## 4.44.3 Member Function Documentation

### 4.44.3.1 a()

```
const glm::vec3 & TesselationPatch::a ( ) const
```

Definition at line 30 of file TessellationPatch.cpp.

#### 4.44.3.2 b()

```
const glm::vec3 & TesselationPatch::b ( ) const
```

Definition at line 35 of file TessellationPatch.cpp.

#### 4.44.3.3 c()

```
const glm::vec3 & TesselationPatch::c ( ) const
```

Definition at line 40 of file TessellationPatch.cpp.

#### 4.44.3.4 move()

```
void TesselationPatch::move (
            const glm::vec3 & delta )
```

Definition at line 21 of file TessellationPatch.cpp.

#### 4.44.3.5 set_abc()

```
void TesselationPatch::set_abc (
            const glm::vec3 & A,
            const glm::vec3 & B,
            const glm::vec3 & C )
```

Definition at line 16 of file TessellationPatch.cpp.

#### 4.44.3.6 vertices()

```
std::vector< glm::vec3 > TesselationPatch::vertices ( )
```

Definition at line 45 of file TessellationPatch.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/TessellationPatch.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/TessellationPatch.cpp

## 4.45 Tessellator Class Reference

`#include <Tessellator.hpp>`

### Static Public Member Functions

- static bool tesselate (std::vector< glm::vec3 > &out, const std::vector< glm::vec3 > &vertices)

### 4.45.1 Detailed Description

Definition at line 15 of file Tessellator.hpp.

### 4.45.2 Member Function Documentation

#### 4.45.2.1 tesselate()

```
bool Tessellator::tesselate (
            std::vector< glm::vec3 > & out,
            const std::vector< glm::vec3 > & vertices )  [static]
```

Definition at line 11 of file Tessellator.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Tessellator.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Tessellator.cpp

## 4.46 TextureCache Class Reference

`#include <TextureCache.hpp>`

### Public Member Functions

- TextureCache ()
- ~TextureCache ()
- GLTexture texture (const std::string &texture_path)

### 4.46.1 Detailed Description

Definition at line 9 of file TextureCache.hpp.

### 4.46.2 Constructor & Destructor Documentation

#### 4.46.2.1 TextureCache()

```
TextureCache::TextureCache ( )
```

Definition at line 6 of file TextureCache.cpp.

#### 4.46.2.2 ~TextureCache()

```
TextureCache::~TextureCache ( )
```

Definition at line 11 of file TextureCache.cpp.

### 4.46.3 Member Function Documentation

#### 4.46.3.1 texture()

```
GLTexture TextureCache::texture (
            const std::string & texture_path )
```

Definition at line 16 of file TextureCache.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/TextureCache.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/TextureCache.cpp

## 4.47 Time Class Reference

```
#include <Timing.hpp>
```

### Static Public Member Functions

- static double get_time ()
- static void set_time_delta (double td)
- static double time_delta ()

### 4.47.1 Detailed Description

Definition at line 8 of file Timing.hpp.

### 4.47.2 Member Function Documentation

#### 4.47.2.1 get_time()

```
static double Time::get_time ( )  [inline], [static]
```

Definition at line 11 of file Timing.hpp.

#### 4.47.2.2 set_time_delta()

```
static void Time::set_time_delta (
            double td )  [inline], [static]
```

Definition at line 18 of file Timing.hpp.

#### 4.47.2.3 time_delta()

```
static double Time::time_delta ( )  [inline], [static]
```

Definition at line 23 of file Timing.hpp.

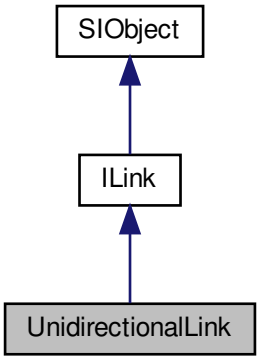
The documentation for this class was generated from the following file:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Timing.hpp

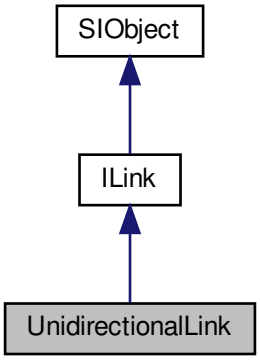
## 4.48 UnidirectionalLink Class Reference

`#include <Link.hpp>`

Inheritance diagram for UnidirectionalLink:




Collaboration diagram for UnidirectionalLink:




### Public Member Functions

- UnidirectionalLink (const std::shared_ptr< Region > &ra, const std::shared_ptr< Region > &rb, const std::string &aa, const std::string &ab)
- ~UnidirectionalLink ()
- const LINK_TYPE & type () const override
- const std::shared_ptr< Region > & sender_a () const override

- const std::shared_ptr< Region > & sender_b () const override
- const std::shared_ptr< Region > & receiver_a () const override
- const std::shared_ptr< Region > & receiver_b () const override
- const std::string & attribute_a () const override
- const std::string & attribute_b () const override
- virtual void add_child (std::shared_ptr< ILink > &link) override
- std::vector< std::shared_ptr< ILink > > & children () override

### Additional Inherited Members

## 4.48.1 Detailed Description

Definition at line 40 of file Link.hpp.

## 4.48.2 Constructor & Destructor Documentation

### 4.48.2.1 UnidirectionalLink()

```
UnidirectionalLink::UnidirectionalLink (
            const std::shared_ptr< Region > & ra,
            const std::shared_ptr< Region > & rb,
            const std::string & aa,
            const std::string & ab )
```

Definition at line 5 of file Link.cpp.

### 4.48.2.2 ~UnidirectionalLink()

```
UnidirectionalLink::~UnidirectionalLink ( )
```

Definition at line 18 of file Link.cpp.

## 4.48.3 Member Function Documentation

### 4.48.3.1 add_child()

```
void UnidirectionalLink::add_child (
            std::shared_ptr< ILink > & link )  [override], [virtual]
```

Implements ILink.

Definition at line 121 of file Link.cpp.

#### 4.48.3.2 attribute_a()

```
const std::string & UnidirectionalLink::attribute_a ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 48 of file Link.cpp.

#### 4.48.3.3 attribute_b()

```
const std::string & UnidirectionalLink::attribute_b ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 53 of file Link.cpp.

#### 4.48.3.4 children()

```
std::vector< std::shared_ptr< ILink > > & UnidirectionalLink::children ( )  [override], [virtual]
```

Implements ILink.

Definition at line 111 of file Link.cpp.

#### 4.48.3.5 receiver_a()

```
const std::shared_ptr< Region > & UnidirectionalLink::receiver_a ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 38 of file Link.cpp.

#### 4.48.3.6 receiver_b()

```
const std::shared_ptr< Region > & UnidirectionalLink::receiver_b ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 43 of file Link.cpp.

#### 4.48.3.7 sender_a()

```
const std::shared_ptr< Region > & UnidirectionalLink::sender_a ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 28 of file Link.cpp.

#### 4.48.3.8 sender_b()

```
const std::shared_ptr< Region > & UnidirectionalLink::sender_b ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 33 of file Link.cpp.

#### 4.48.3.9 type()

```
const ILink::LINK_TYPE & UnidirectionalLink::type ( ) const  [override], [virtual]
```

Implements ILink.

Definition at line 23 of file Link.cpp.

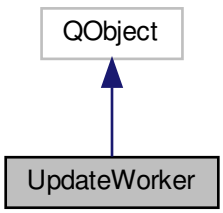
The documentation for this class was generated from the following files:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linking/Link.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linking/Link.cpp

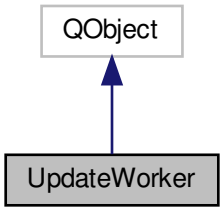
## 4.49 UpdateWorker Class Reference

```
#include <UpdateWorker.hpp>
```

Inheritance diagram for UpdateWorker:



Collaboration diagram for UpdateWorker:




### Public Member Functions

- UpdateWorker ()
- ~UpdateWorker ()
- void start ()
- Q_SLOT void stop ()
- void pause ()
- void resume ()
- bool is_running ()
- int fps () const
- Q_SIGNAL void running_changed ()
- Q_SIGNAL void updated (double delta, int fps)
- Q_SIGNAL void finished ()

## 4.49.1 Detailed Description

Definition at line 10 of file UpdateWorker.hpp.

## 4.49.2 Constructor & Destructor Documentation

### 4.49.2.1 UpdateWorker()

```
UpdateWorker::UpdateWorker ( )  [explicit]
```

Definition at line 5 of file UpdateWorker.cpp.

#### 4.49.2.2 ~UpdateWorker()

```
UpdateWorker::~UpdateWorker ( )
```

Definition at line 11 of file UpdateWorker.cpp.

### 4.49.3 Member Function Documentation

#### 4.49.3.1 finished()

```
Q_SIGNAL void UpdateWorker::finished ( )
```

#### 4.49.3.2 fps()

```
int UpdateWorker::fps ( ) const
```

Definition at line 70 of file UpdateWorker.cpp.

#### 4.49.3.3 is_running()

```
bool UpdateWorker::is_running ( )
```

Definition at line 65 of file UpdateWorker.cpp.

#### 4.49.3.4 pause()

```
void UpdateWorker::pause ( )
```

Definition at line 44 of file UpdateWorker.cpp.

#### 4.49.3.5 resume()

```
void UpdateWorker::resume ( )
```

Definition at line 53 of file UpdateWorker.cpp.

#### 4.49.3.6 running_changed()

```
Q_SIGNAL void UpdateWorker::running_changed ( )
```

#### 4.49.3.7 start()

```
void UpdateWorker::start ( )
```

Definition at line 21 of file UpdateWorker.cpp.

#### 4.49.3.8 stop()

```
void UpdateWorker::stop ( )
```

Definition at line 30 of file UpdateWorker.cpp.

#### 4.49.3.9 updated()

```
Q_SIGNAL void UpdateWorker::updated (
            double delta,
            int fps )
```

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/background/UpdateWorker.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/background/UpdateWorker.cpp

## 4.50 UUID Class Reference

```
#include <UUID.hpp>
```

### Static Public Member Functions

- static std::string uuid ()

### 4.50.1 Detailed Description

Definition at line 9 of file UUID.hpp.

### 4.50.2 Member Function Documentation

#### 4.50.2.1 uuid()

```
static std::string UUID::uuid ( )  [inline], [static]
```

Definition at line 12 of file UUID.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/util/UUID.hpp

## 4.51 UV Struct Reference

```
#include <Vertex.hpp>
```

### Public Attributes

- float u
- float v

### 4.51.1 Detailed Description

Definition at line 30 of file Vertex.hpp.

### 4.51.2 Member Data Documentation

#### 4.51.2.1 u

```
float UV::u
```

Definition at line 32 of file Vertex.hpp.

#### 4.51.2.2 v

```
float UV::v
```

Definition at line 33 of file Vertex.hpp.

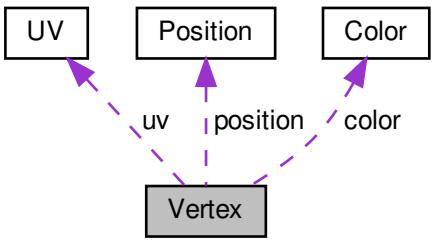
The documentation for this struct was generated from the following file:

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Vertex.hpp

## 4.52 Vertex Struct Reference

```
#include <Vertex.hpp>
```

Collaboration diagram for Vertex:




### Public Member Functions

- void set_position (float x, float y)
- void set_uv (float u, float v)
- void set_color (GLubyte r, GLubyte g, GLubyte b, GLubyte a)

### Public Attributes

- struct Position position
- struct Color color
- struct UV uv

### 4.52.1 Detailed Description

Definition at line 36 of file Vertex.hpp.

### 4.52.2 Member Function Documentation

#### 4.52.2.1 set_color()

```
void Vertex::set_color (
            GLubyte r,
            GLubyte g,
            GLubyte b,
            GLubyte a )  [inline]
```

Definition at line 54 of file Vertex.hpp.

#### 4.52.2.2 set_position()

```
void Vertex::set_position (
            float x,
            float y )  [inline]
```

Definition at line 42 of file Vertex.hpp.

#### 4.52.2.3 set_uv()

```
void Vertex::set_uv (
            float u,
            float v )  [inline]
```

Definition at line 48 of file Vertex.hpp.

### 4.52.3 Member Data Documentation

#### 4.52.3.1 color

```
struct Color Vertex::color
```

Definition at line 39 of file Vertex.hpp.

#### 4.52.3.2 position

```
struct Position Vertex::position
```

Definition at line 38 of file Vertex.hpp.

#### 4.52.3.3 uv

```
struct UV Vertex::uv
```

Definition at line 40 of file Vertex.hpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Vertex.hpp

## 4.53 Window Class Reference

```
#include <Window.hpp>
```

### Public Member Functions

- Window ()
- ~Window ()
- int create (const std::string &window_name, int width, int height, unsigned int flags)
- void swap_buffer ()
- int width () const
- void set_width (int width)
- int height () const
- void set_height (int height)

### 4.53.1 Detailed Description

Definition at line 18 of file Window.hpp.

### 4.53.2 Constructor & Destructor Documentation

#### 4.53.2.1 Window()

```
Window::Window ( )
```

Definition at line 9 of file Window.cpp.

#### 4.53.2.2 ~Window()

```
Window::~Window ( )
```

Definition at line 12 of file Window.cpp.

### 4.53.3 Member Function Documentation

#### 4.53.3.1 create()

```
int Window::create (
             const std::string & window_name,
             int width,
             int height,
             unsigned int flags )
```

Definition at line 17 of file Window.cpp.

#### 4.53.3.2 height()

```
int Window::height ( ) const
```

Definition at line 67 of file Window.cpp.

#### 4.53.3.3 set_height()

```
void Window::set_height (
             int height )
```

Definition at line 72 of file Window.cpp.

#### 4.53.3.4 set_width()

```
void Window::set_width (
             int width )
```

Definition at line 62 of file Window.cpp.

#### 4.53.3.5 swap_buffer()

```
void Window::swap_buffer ( )
```

Definition at line 52 of file Window.cpp.

#### 4.53.3.6 width()

```
int Window::width ( ) const
```

Definition at line 57 of file Window.cpp.

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

- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Window.hpp
- /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Window.cpp

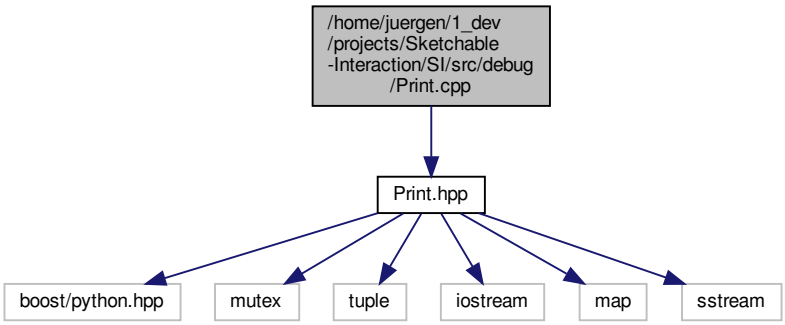
# Chapter 5

# File Documentation

## 5.1 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/debug/Print.cpp File Reference

```
#include "Print.hpp"
```

Include dependency graph for Print.cpp:

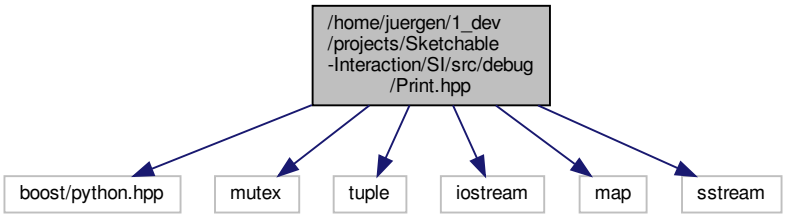



## 5.2 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/debug/Print.hpp File Reference

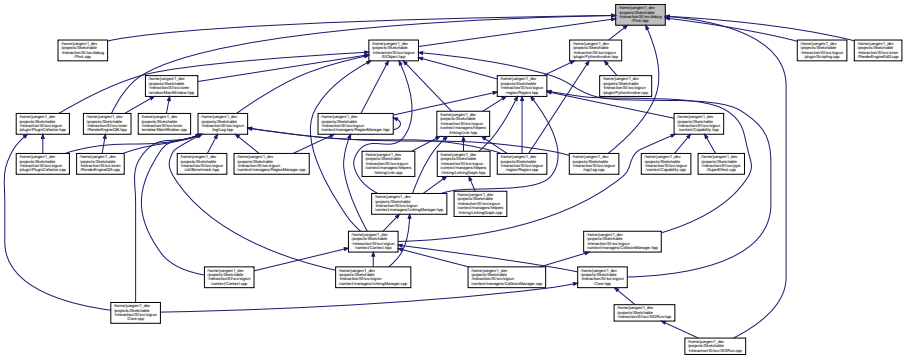
```
#include <boost/python.hpp>
#include <mutex>
#include <tuple>
#include <iostream>
#include <map>
```

```
#include <sstream>
```

Include dependency graph for Print.hpp:




This graph shows which files directly or indirectly include this file:



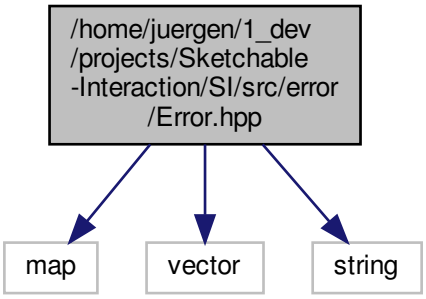
## Classes

- class Print

# 5.3 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/error/Error.hpp File Reference

```
#include <map>
#include <vector>
#include <string>
```

Include dependency graph for Error.hpp:




## Macros

- #define EN 0
- #define DE 1
- #define ERROR_SIGRUN 100
- #define ERROR_IO 200
- #define ERROR_PYTHON 300
- #define ERROR_UNKNOWN 900
- #define ERRORS_EN
- #define ERRORS_DE

## Variables

- int LANGUAGE = 0
- std::map< int, std::map< int, std::string > > ERRORS

### 5.3.1 Macro Definition Documentation

#### 5.3.1.1 DE

```
#define DE 1
```

Definition at line 10 of file Error.hpp.

#### 5.3.1.2 EN

```
#define EN 0
```

Definition at line 9 of file Error.hpp.

#### 5.3.1.3 ERROR_IO

```
#define ERROR_IO 200
```

Definition at line 13 of file Error.hpp.

#### 5.3.1.4 ERROR_PYTHON

```
#define ERROR_PYTHON 300
```

Definition at line 14 of file Error.hpp.

#### 5.3.1.5 ERROR_SIGRUN

```
#define ERROR_SIGRUN 100
```

Definition at line 12 of file Error.hpp.

#### 5.3.1.6 ERROR_UNKNOWN

```
#define ERROR_UNKNOWN 900
```

Definition at line 16 of file Error.hpp.

#### 5.3.1.7 ERRORS_DE

```
#define ERRORS_DE
```

**Value:**

```
{\
    {ERROR_PYTHON, "In einem Python-Plugin ist ein Fehler aufgetreten."},\
    {ERROR_IO, "Bei Input/Output ist ein Fehler aufgetreten."}\
}
```

Definition at line 24 of file Error.hpp.

#### 5.3.1.8 ERRORS_EN

```
#define ERRORS_EN
```

**Value:**

```
{\
    {ERROR_PYTHON, "An error with a python plugin occured."},\
    {ERROR_IO, "An error with input/output occured."}\
}
```

Definition at line 18 of file Error.hpp.

### 5.3.2 Variable Documentation

#### 5.3.2.1 ERRORS

```
std::map<int, std::map<int, std::string> > ERRORS
```

**Initial value:**

```
{
        {EN, ERRORS_EN},
        {DE, ERRORS_DE}
}
```

Definition at line 33 of file Error.hpp.

#### 5.3.2.2 LANGUAGE

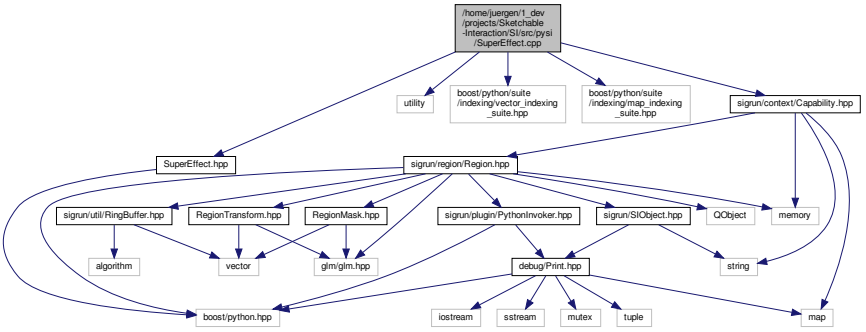
```
int LANGUAGE = 0
```

Definition at line 30 of file Error.hpp.

## 5.4 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/pysi/SuperEffect.cpp File Reference

```
#include "SuperEffect.hpp"
#include <utility>
#include <boost/python/suite/indexing/vector_indexing_suite.hpp>
#include <boost/python/suite/indexing/map_indexing_suite.hpp>
#include <sigrun/context/Capability.hpp>
```

Include dependency graph for SuperEffect.cpp:



### Functions

- BOOST_PYTHON_MODULE (libPySI)

### 5.4.1 Function Documentation

#### 5.4.1.1 BOOST_PYTHON_MODULE()

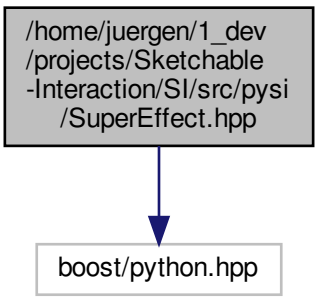
```
BOOST_PYTHON_MODULE (
             libPySI  )
```

Definition at line 63 of file SuperEffect.cpp.

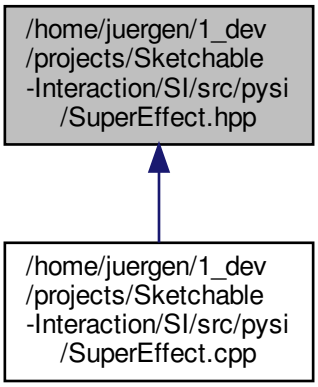
## 5.5 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/pysi/SuperEffect.hpp File Reference

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

Include dependency graph for SuperEffect.hpp:



This graph shows which files directly or indirectly include this file:



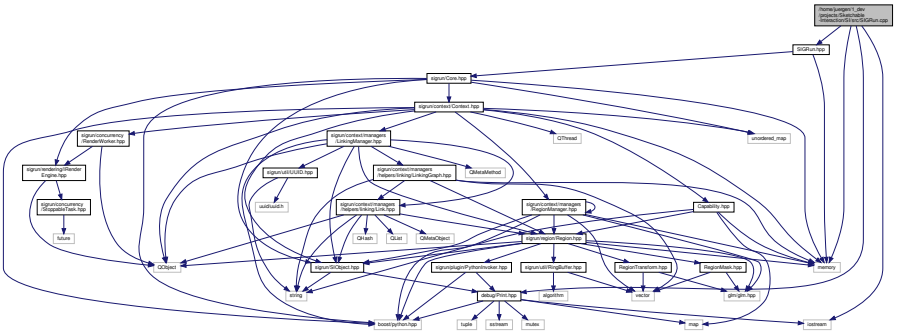

### Classes

- class IterableConverter
- class SuperEffect
- class PySIEffect

## 5.6 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/SIGRun.cpp File Reference

```
#include <memory>
#include <iostream>
#include "SIGRun.hpp"
#include "debug/Print.hpp"
```

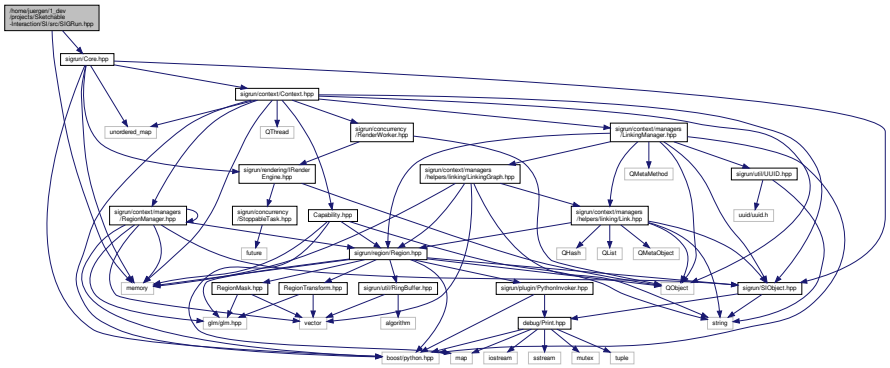
Include dependency graph for SIGRun.cpp:



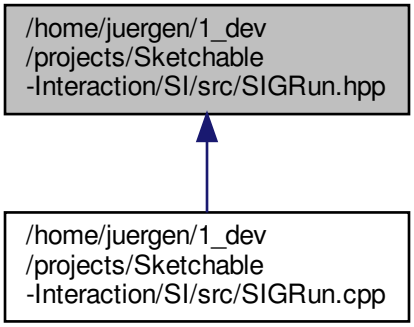
## 5.7 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/SIGRun.hpp File Reference

```
#include <memory>
#include "sigrun/Core.hpp"
```

Include dependency graph for SIGRun.hpp:




This graph shows which files directly or indirectly include this file:




### Classes

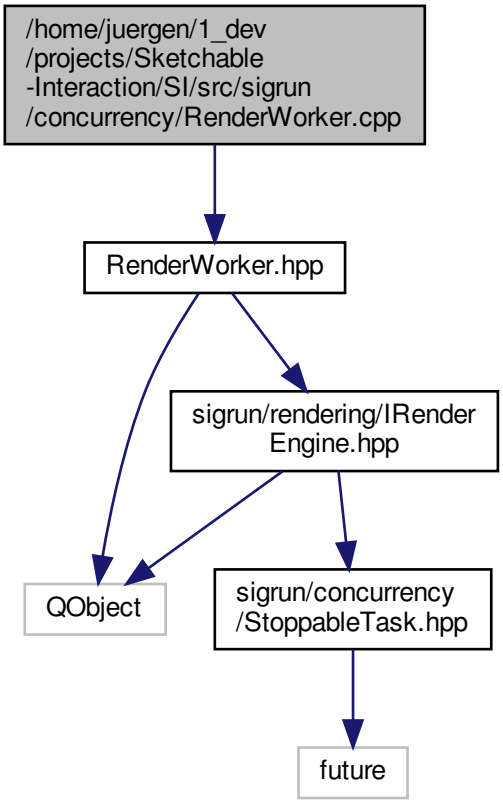
- class SIGRun

  *SIGRun class serving as entry point of an SI environment.*

## 5.8 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/concurrency/RenderWorker.cpp File Reference

```
#include "RenderWorker.hpp"
```

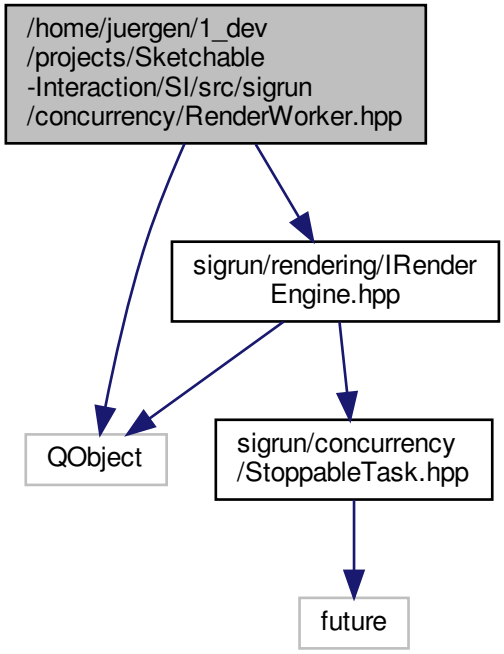
Include dependency graph for RenderWorker.cpp:



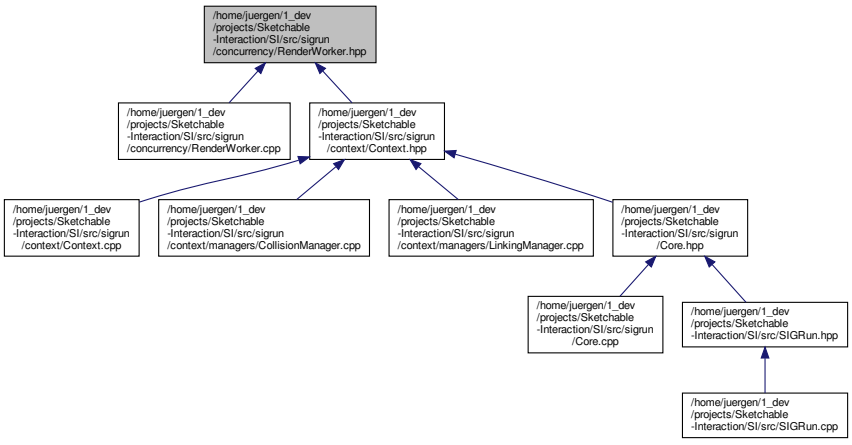

## 5.9 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/concurrency/RenderWorker.hpp File Reference

```
#include <QObject>
#include <sigrun/rendering/IRenderEngine.hpp>
```

Include dependency graph for RenderWorker.hpp:




This graph shows which files directly or indirectly include this file:



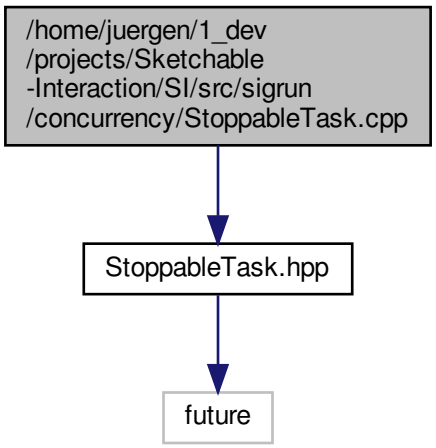

## Classes

- class RenderWorker

## 5.10 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/concurrency/↩ StoppableTask.cpp File Reference

```
#include "StoppableTask.hpp"
```

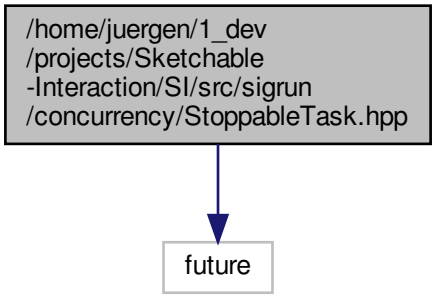
Include dependency graph for StoppableTask.cpp:



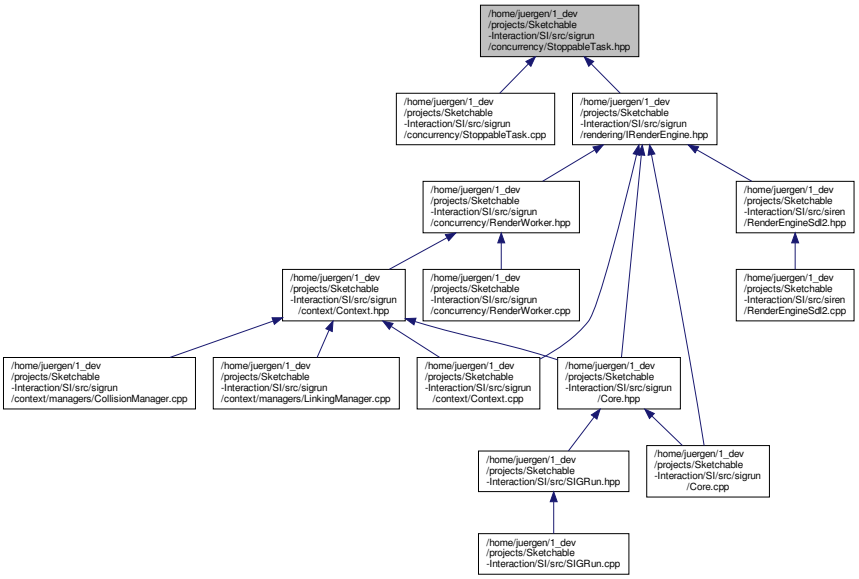

## 5.11 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/concurrency/↩ StoppableTask.hpp File Reference

```
#include <future>
```

Include dependency graph for StoppableTask.hpp:



This graph shows which files directly or indirectly include this file:



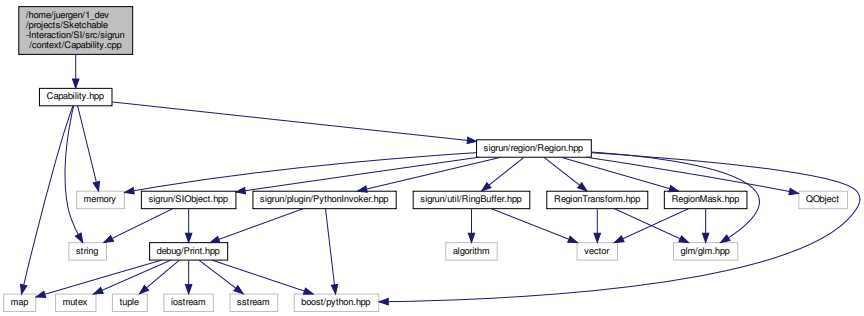

### Classes

- class StoppableTask

## 5.12 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/Capability.cpp File Reference

```
#include "Capability.hpp"
```

Include dependency graph for Capability.cpp:

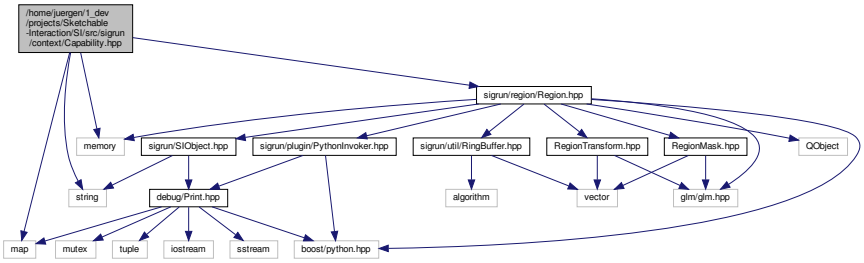



## 5.13 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/Capability.hpp File Reference

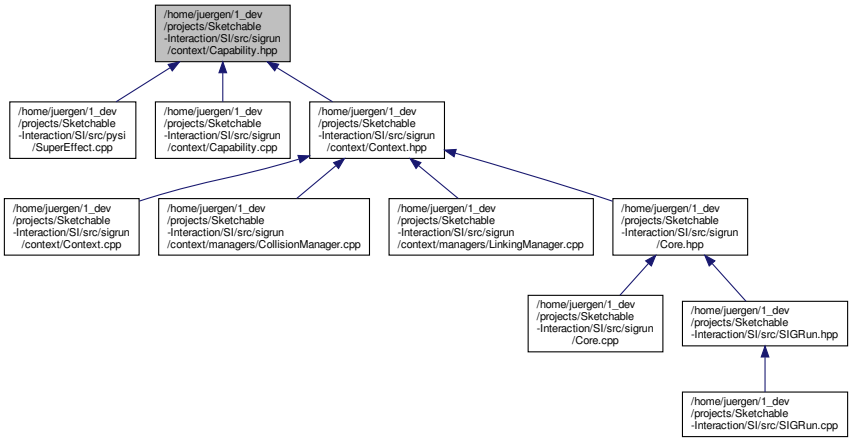
```
#include <string>
#include <map>
```

```
#include <memory>
#include <sigrun/region/Region.hpp>
```

Include dependency graph for Capability.hpp:




This graph shows which files directly or indirectly include this file:




### Classes

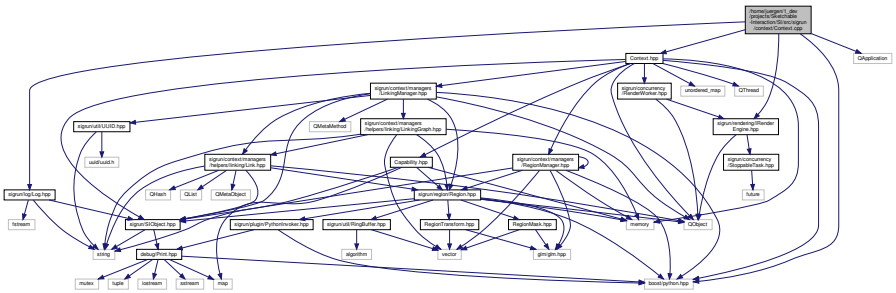
- class Capability

## 5.14 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/Context.cpp File Reference

```
#include <sigrun/log/Log.hpp>
#include "Context.hpp"
#include <QApplication>
#include <sigrun/rendering/IRenderEngine.hpp>
```

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

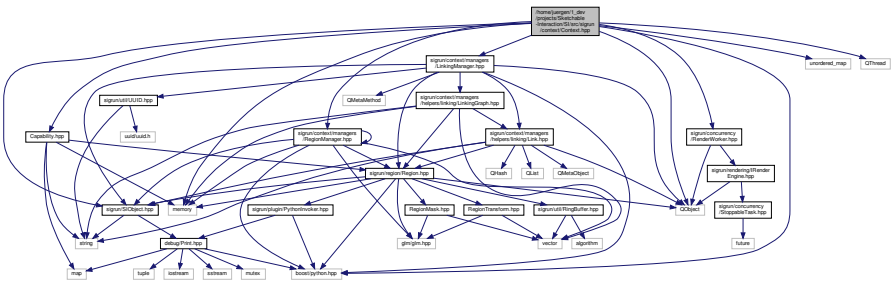
Include dependency graph for Context.cpp:



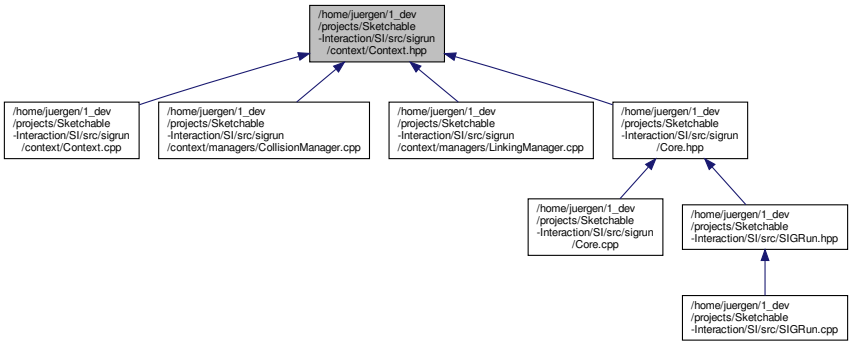
## 5.15 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/Context.hpp File Reference

```
#include <sigrun/SIObject.hpp>
#include "Capability.hpp"
#include <unordered_map>
#include <memory>
#include <boost/python.hpp>
#include <sigrun/context/managers/RegionManager.hpp>
#include <sigrun/context/managers/LinkingManager.hpp>
#include <sigrun/concurrency/RenderWorker.hpp>
#include <QThread>
#include <QObject>
```

Include dependency graph for Context.hpp:



This graph shows which files directly or indirectly include this file:



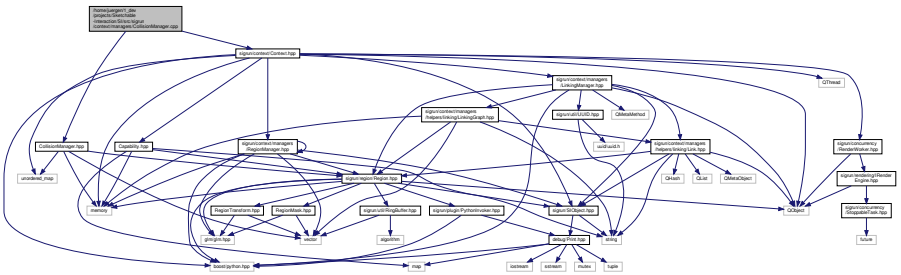

## Classes

- class Context

# 5.16 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/↩ CollisionManager.cpp File Reference

```
#include "CollisionManager.hpp"
#include <sigrun/context/Context.hpp>
```

Include dependency graph for CollisionManager.cpp:

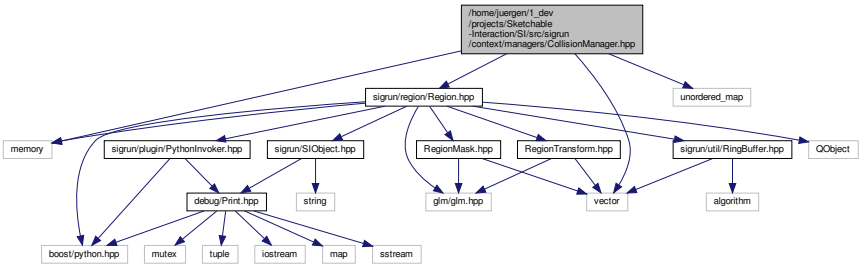


# 5.17 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/↩ CollisionManager.hpp File Reference

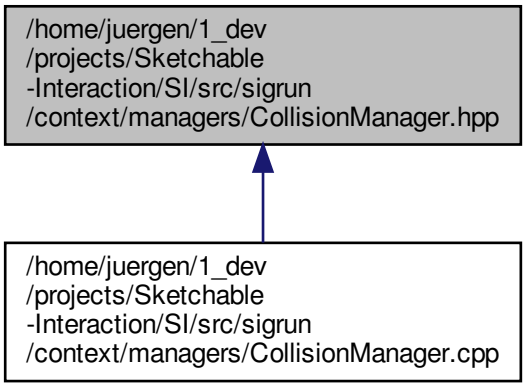
```
#include <memory>
#include <vector>
#include <sigrun/region/Region.hpp>
```

```
#include <unordered_map>
```

Include dependency graph for CollisionManager.hpp:




This graph shows which files directly or indirectly include this file:



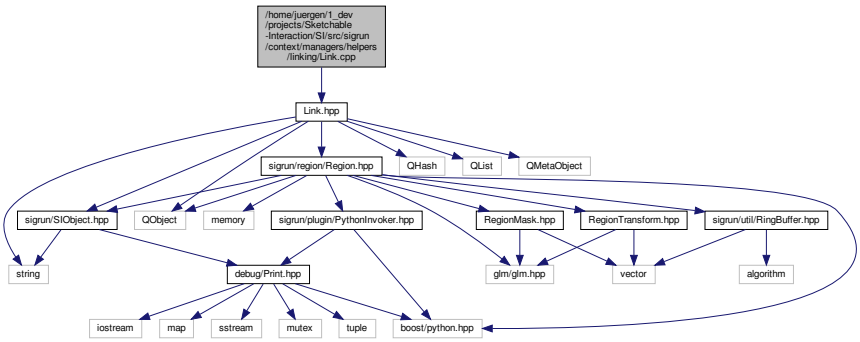

### Classes

- class CollisionManager

## 5.18 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linki Link.cpp File Reference

```
#include "Link.hpp"
```

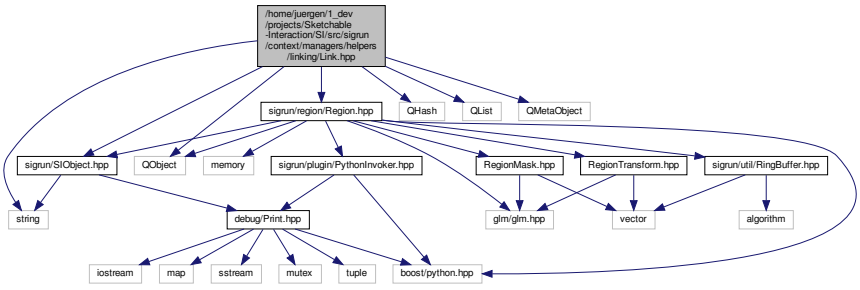
Include dependency graph for Link.cpp:



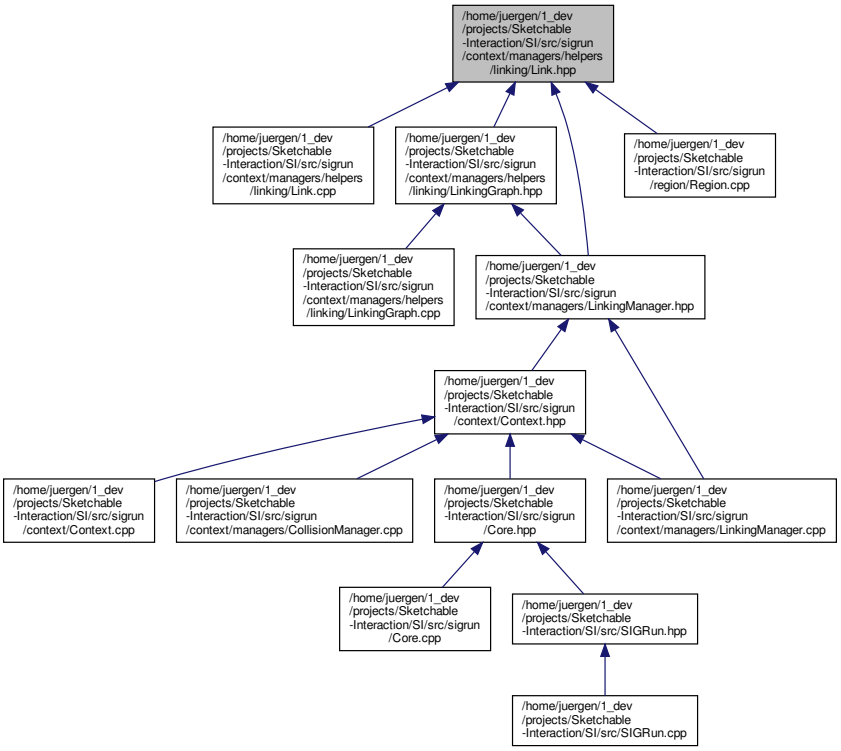

## 5.19 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linki Link.hpp File Reference

```
#include <string>
#include <sigrun/SIObject.hpp>
#include <QHash>
#include <QList>
#include <QMetaObject>
#include <QObject>
#include <sigrun/region/Region.hpp>
```

Include dependency graph for Link.hpp:



This graph shows which files directly or indirectly include this file:



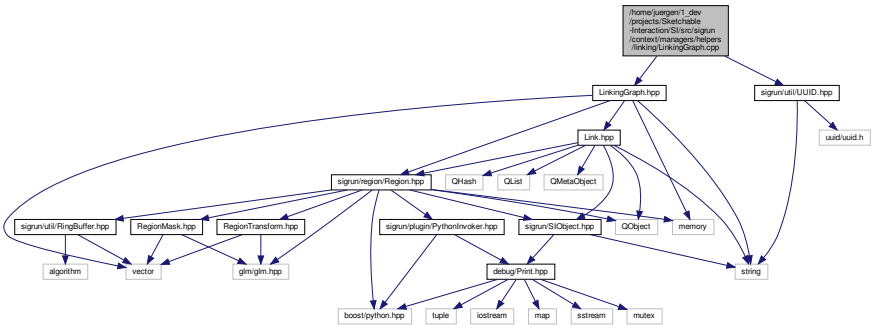

### Classes

- class ILink
- class UnidirectionalLink
- class BidirectionalLink

## 5.20 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linki LinkingGraph.cpp File Reference

```
#include "LinkingGraph.hpp"
#include <sigrun/util/UUID.hpp>
```

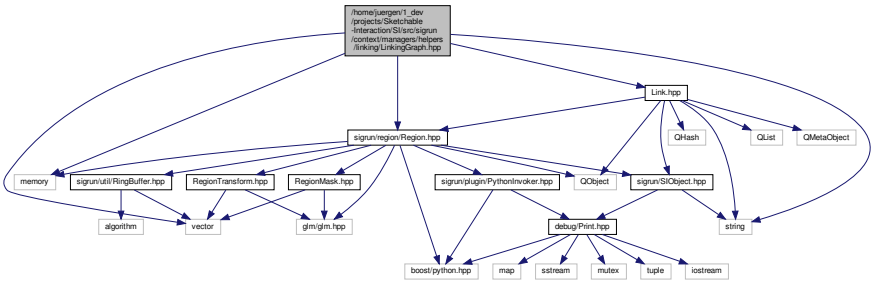
Include dependency graph for LinkingGraph.cpp:



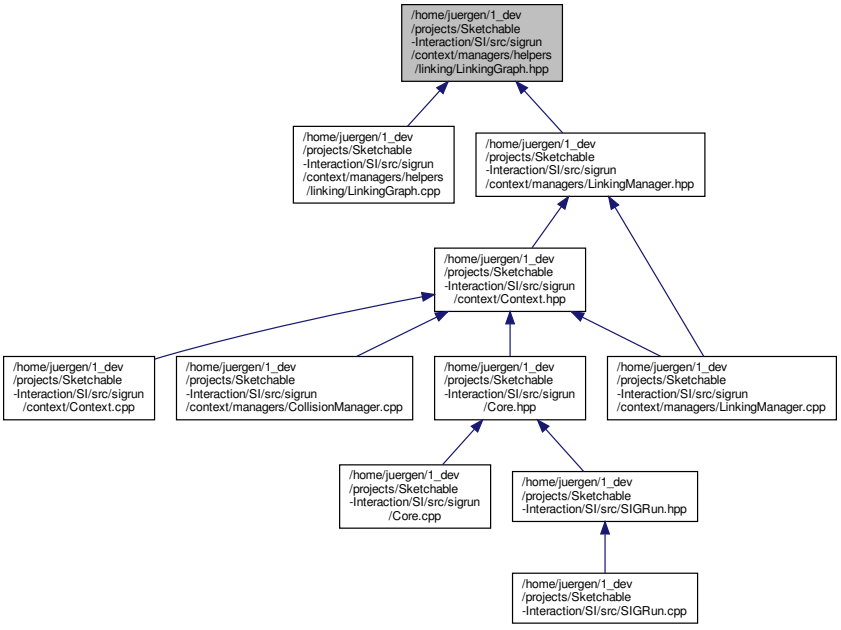
# 5.21 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/helpers/linki LinkingGraph.hpp File Reference

```
#include <memory>
#include <vector>
#include <string>
#include <sigrun/region/Region.hpp>
#include "Link.hpp"
```

Include dependency graph for LinkingGraph.hpp:




This graph shows which files directly or indirectly include this file:



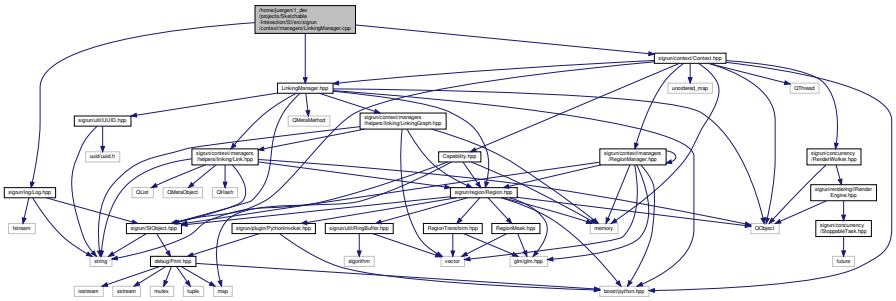

## Classes

- class LinkingGraph

## 5.22 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/LinkingManager.cpp File Reference

```
#include "LinkingManager.hpp"
#include <sigrun/log/Log.hpp>
#include <sigrun/context/Context.hpp>
```

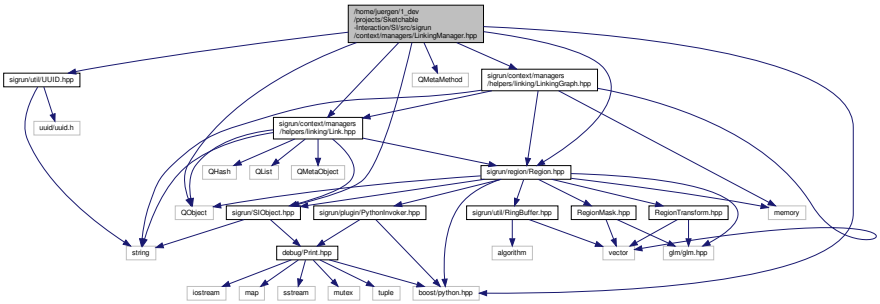
Include dependency graph for LinkingManager.cpp:



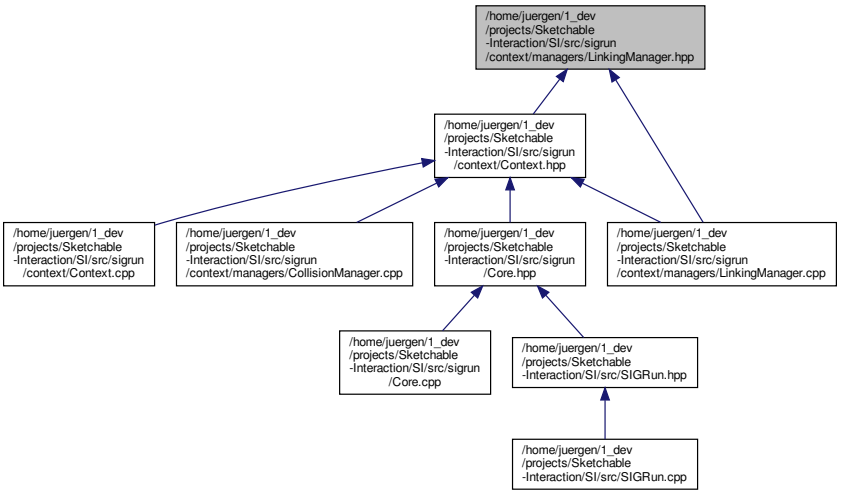

## 5.23 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/LinkingManager.hpp File Reference

```
#include <sigrun/SIObject.hpp>
#include <QObject>
#include <sigrun/region/Region.hpp>
#include <QMetaMethod>
#include <sigrun/util/UUID.hpp>
#include <boost/python.hpp>
#include "sigrun/context/managers/helpers/linking/Link.hpp"
#include "sigrun/context/managers/helpers/linking/LinkingGraph.hpp"
```

Include dependency graph for LinkingManager.hpp:



This graph shows which files directly or indirectly include this file:



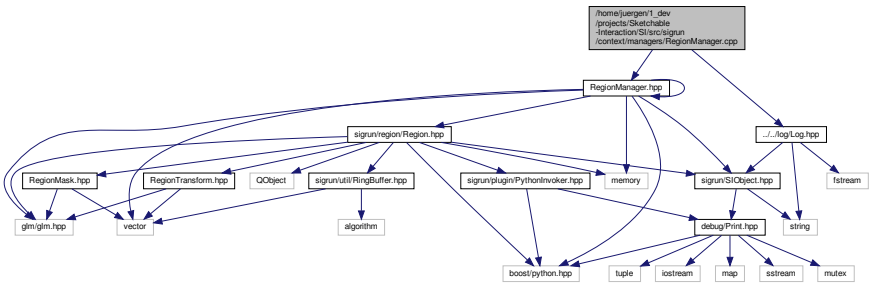

### Classes

- class LinkingManager

## 5.24 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/RegionManager.cpp File Reference

```
#include "RegionManager.hpp"
#include "../../log/Log.hpp"
```

Include dependency graph for RegionManager.cpp:

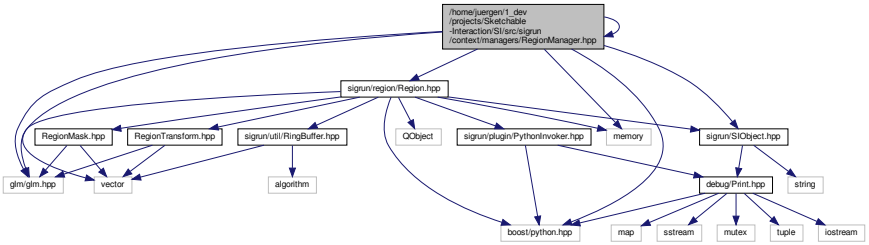



## 5.25 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/context/managers/RegionManager.hpp File Reference

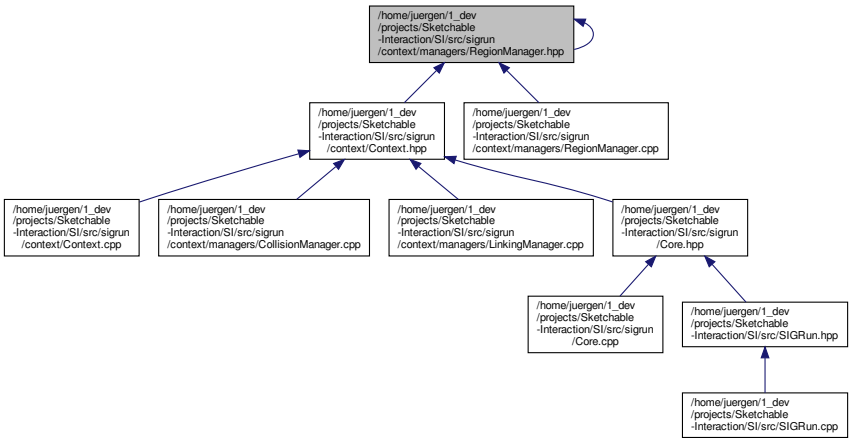
```
#include <vector>
#include <glm/glm.hpp>
```

```
#include <boost/python.hpp>
#include <memory>
#include <sigrun/region/Region.hpp>
#include <sigrun/context/managers/RegionManager.hpp>
#include "../../SIObject.hpp"
```

Include dependency graph for RegionManager.hpp:




This graph shows which files directly or indirectly include this file:




### Classes

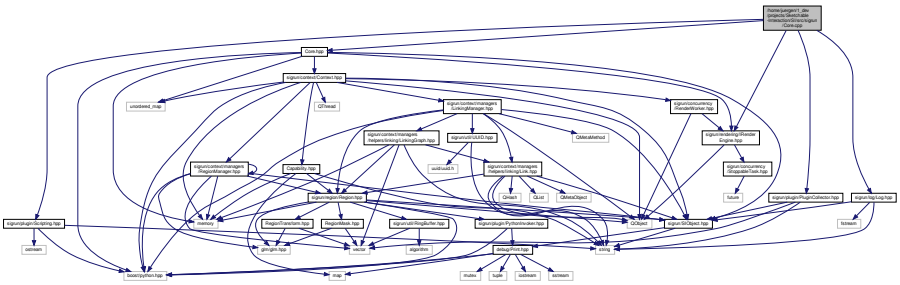
- class RegionManager

## 5.26 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/Core.cpp File Reference

```
#include <sigrun/log/Log.hpp>
#include <sigrun/rendering/IRenderEngine.hpp>
#include "Core.hpp"
#include "sigrun/plugin/Scripting.hpp"
```

```
#include "sigrun/plugin/PluginCollector.hpp"
```

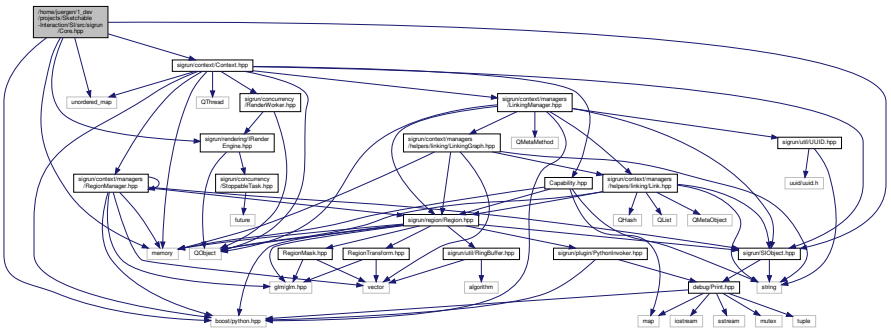
Include dependency graph for Core.cpp:



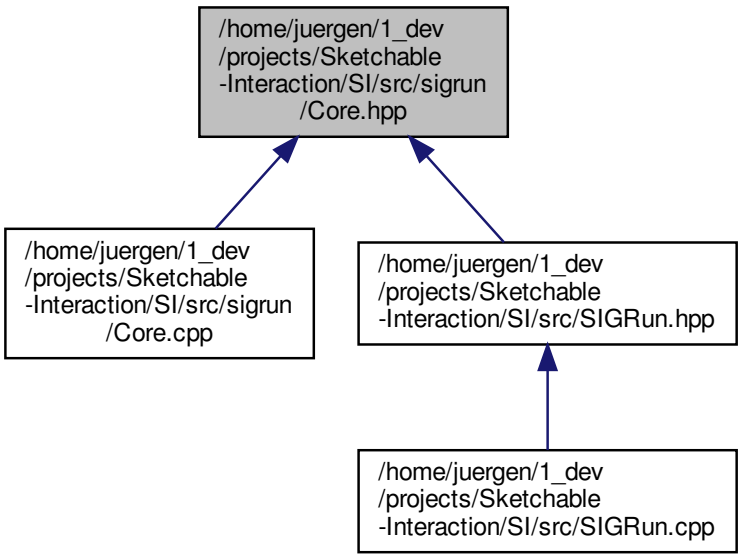
## 5.27 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/Core.hpp File Reference

```
#include <boost/python.hpp>
#include <memory>
#include <unordered_map>
#include <sigrun/rendering/IRenderEngine.hpp>
#include <sigrun/context/Context.hpp>
#include "SIObject.hpp"
```

Include dependency graph for Core.hpp:



This graph shows which files directly or indirectly include this file:




### Classes

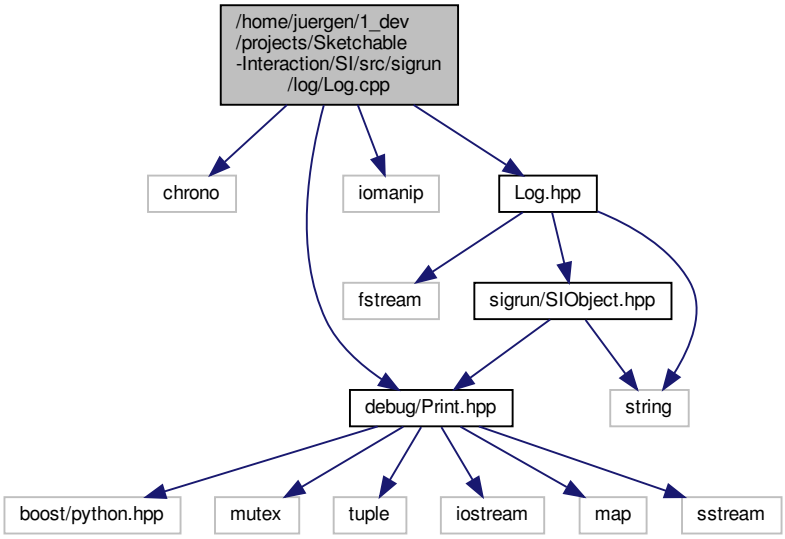
- class Core

  *namespace shortening for python object integration*

## 5.28 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/log/Log.cpp File Reference

```
#include <chrono>
#include <debug/Print.hpp>
#include <iomanip>
#include "Log.hpp"
```

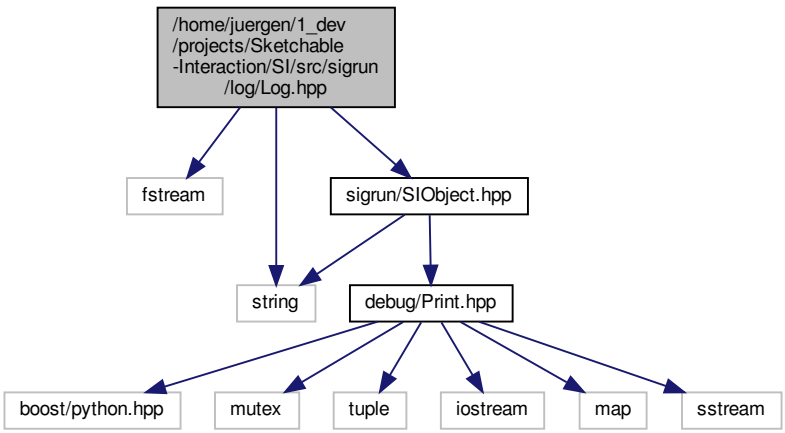
Include dependency graph for Log.cpp:



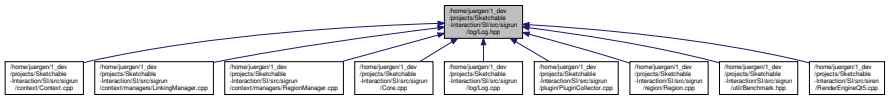

## 5.29 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/log/Log.hpp File Reference

```
#include <fstream>
#include <string>
#include "sigrun/SIObject.hpp"
```

Include dependency graph for Log.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class Log

  *Log class serving as central logging functionality for easy logging data output.*

## Macros

- #define ERROR_COLOR(x) ("\033[31m" + x + "\033[0m")

  *red coloring for console output*
- #define UNDEFINED_COLOR(x) ("\033[1;31m" + x + "\033[0m")

  *bold red coloring for console output*
- #define INFO_COLOR(x) ("\033[32m" + x + "\033[0m")

  *green coloring for console output*
- #define WARN_COLOR(x) ("\033[33m" + x + "\033[0m")

  *yellow coloring for console output*
- #define DEBUG_COLOR(x) ("\033[37m" + x + "\033[0m")

  *white/gray coloring for console output*
- #define __FILENAME__ (strrchr(__FILE__, '/') ? strrchr(__FILE__, '/') + 1 : __FILE__)

  *file name and extension without full path*
- #define LOG_NONE Log::MODE::NONE

  *disable logging output*
- #define LOG_CONSOLE Log::MODE::CONSOLE

  *output logging data to stdout*
- #define LOG_FILE Log::MODE::FILE

  *output logging data to file*
- #define LOG_SHOW_NONE Log::SHOW_TYPE::HIDDEN

  *disable logging except for errors and undefined behaviour*
- #define LOG_SHOW_INFO Log::SHOW_TYPE::INFO

  *enable logging of data tagged as INFO (information) additionally to errors and undefined behaviour*
- #define LOG_SHOW_WARN Log::SHOW_TYPE::WARN

  *enable logging of data tagged as WARN (warning) additionally to errors and undefined behaviour*
- #define LOG_SHOW_ERROR Log::SHOW_TYPE::ERROR

  *enable logging of data tagged as ERROR (error) however this per default enabled and cannot be disabled*
- #define LOG_SHOW_DEBUG Log::SHOW_TYPE::DEBUG

  *enable logging of data tagged as DEBUG (debugging information) additionally to errors and undefined behaviour*
- #define LOG_SHOW_ALL Log::SHOW_TYPE::INFO | Log::SHOW_TYPE::WARN | Log::SHOW_TYPE::ERROR | Log::SHOW_TYPE::DEBUG

  *enable logging of any tagged data*
- #define DEBUG(what) Log::log(origin(), what, Log::LOG_LEVEL::DEBUG_LEVEL, meta_type(),__FILENAME__, __FUNCTION__, std::to_string(__LINE__))

  *perform logging of data with the DEBUG tag*
- #define INFO(what) Log::log(origin(), what, Log::LOG_LEVEL::INFO_LEVEL, meta_type())

*perform logging of data with the INFO tag*

- #define ERROR(what) Log::log(origin(), what, Log::LOG_LEVEL::ERROR_LEVEL, meta_type(),__FILENAME__, __FUNCTION__, std::to_string(__LINE__))

  *perform logging of data with the ERROR tag*
- #define WARN(what) Log::log(origin(), what, Log::LOG_LEVEL::WARN_LEVEL, meta_type())

  *perform logging of data with the WARN tag*
- #define UNDEFINED(what) Log::log(origin(), what, Log::LOG_LEVEL::UNDEFINED_LEVEL, meta_type(), __FILENAME__, __FUNCTION__, std::to_string(__LINE__))

  *perform logging of data with the UNDEFINED tag*

## 5.29.1 Macro Definition Documentation

### 5.29.1.1 __FILENAME__

```
#define __FILENAME__ (strrchr(__FILE__, '/') ?  strrchr(__FILE__, '/') + 1 :  __FILE__)
```

file name and extension without full path

file name and extension without full path

Definition at line 61 of file Log.hpp.

### 5.29.1.2 DEBUG

```
#define DEBUG(
              what ) Log::log(origin(), what, Log::LOG_LEVEL::DEBUG_LEVEL, meta_type(),__FILENAME__,
__FUNCTION__, std::to_string(__LINE__))
```

perform logging of data with the DEBUG tag

Shortcut macro for logging of data with the DEBUG tag which uses static access of log() function of Log class

**Parameters**

| | |
|---|---|
| *what* | the message to be logged |
| *log_mode* | the description where the message is outputted ( |

**See also**

Log::MODE::CONSOLE or
Log::MODE::FILE or both)
Log::log()

Definition at line 118 of file Log.hpp.

#### 5.29.1.3 DEBUG_COLOR

```
#define DEBUG_COLOR(
            x ) ("\033[37m" + x + "\033[0m")
```

white/gray coloring for console output

Coloring for console output. Unused for file output. See table of codes here: https://en.wikipedia.org/wiki/ANSI_escape_code#graphics

Definition at line 54 of file Log.hpp.

#### 5.29.1.4 ERROR

```
#define ERROR(
            what ) Log::log(origin(), what, Log::LOG_LEVEL::ERROR_LEVEL, meta_type(),__FILENAME__,
__FUNCTION__, std::to_string(__LINE__))
```

perform logging of data with the ERROR tag

Shortcut macro for logging of data with the ERROR tag which uses static access of log() function of Log class

**Parameters**

| | |
|---|---|
| *what* | the message to be logged |
| *log_mode* | the description where the message is outputted ( |

**See also**

Log::MODE::CONSOLE or
Log::MODE::FILE or both)
Log::log()

Definition at line 142 of file Log.hpp.

#### 5.29.1.5 ERROR_COLOR

```
#define ERROR_COLOR(
            x ) ("\033[31m" + x + "\033[0m")
```

red coloring for console output

Coloring for console output. Unused for file output. See table of codes here: https://en.wikipedia.org/wiki/ANSI_escape_code#graphics

Definition at line 18 of file Log.hpp.

#### 5.29.1.6 INFO

```
#define INFO(
              what ) Log::log(origin(), what, Log::LOG_LEVEL::INFO_LEVEL, meta_type())
```

perform logging of data with the INFO tag

Shortcut macro for logging of data with the INFO tag which uses static access of log() function of Log class

**Parameters**

| | |
|---|---|
| *what* | the message to be logged |
| *log_mode* | the description where the message is outputted ( |

**See also**

Log::MODE::CONSOLE or
Log::MODE::FILE or both)
Log::log()

Definition at line 130 of file Log.hpp.

#### 5.29.1.7 INFO_COLOR

```
#define INFO_COLOR(
              x ) ("\033[32m" + x + "\033[0m")
```

green coloring for console output

Coloring for console output. Unused for file output. See table of codes here: https://en.wikipedia.org/wiki/ANSI_escape_code#graphics

Definition at line 36 of file Log.hpp.

#### 5.29.1.8 LOG_CONSOLE

```
#define LOG_CONSOLE Log::MODE::CONSOLE
```

output logging data to stdout

Definition at line 71 of file Log.hpp.

#### 5.29.1.9 LOG_FILE

```
#define LOG_FILE Log::MODE::FILE
```

output logging data to file

Definition at line 76 of file Log.hpp.

#### 5.29.1.10 LOG_NONE

```
#define LOG_NONE Log::MODE::NONE
```

disable logging output

Definition at line 66 of file Log.hpp.

#### 5.29.1.11 LOG_SHOW_ALL

```
#define LOG_SHOW_ALL Log::SHOW_TYPE::INFO | Log::SHOW_TYPE::WARN | Log::SHOW_TYPE::ERROR | Log::SHOW_TYPE::DEBU
```

enable logging of any tagged data

Definition at line 106 of file Log.hpp.

#### 5.29.1.12 LOG_SHOW_DEBUG

```
#define LOG_SHOW_DEBUG Log::SHOW_TYPE::DEBUG
```

enable logging of data tagged as DEBUG (debugging information) additionally to errors and undefined behaviour

Definition at line 101 of file Log.hpp.

#### 5.29.1.13 LOG_SHOW_ERROR

```
#define LOG_SHOW_ERROR Log::SHOW_TYPE::ERROR
```

enable logging of data tagged as ERROR (error) however this per default enabled and cannot be disabled

Definition at line 96 of file Log.hpp.

#### 5.29.1.14 LOG_SHOW_INFO

```
#define LOG_SHOW_INFO Log::SHOW_TYPE::INFO
```

enable logging of data tagged as INFO (information) additionally to errors and undefined behaviour

Definition at line 86 of file Log.hpp.

#### 5.29.1.15 LOG_SHOW_NONE

```
#define LOG_SHOW_NONE Log::SHOW_TYPE::HIDDEN
```

disable logging except for errors and undefined behaviour

Definition at line 81 of file Log.hpp.

#### 5.29.1.16 LOG_SHOW_WARN

```
#define LOG_SHOW_WARN Log::SHOW_TYPE::WARN
```

enable logging of data tagged as WARN (warning) additionally to errors and undefined behaviour

Definition at line 91 of file Log.hpp.

#### 5.29.1.17 UNDEFINED

```
#define UNDEFINED(
              what ) Log::log(origin(), what, Log::LOG_LEVEL::UNDEFINED_LEVEL, meta_type(),
__FILENAME__, __FUNCTION__, std::to_string(__LINE__))
```

perform logging of data with the UNDEFINED tag

Shortcut macro for logging of data with the UNDEFINED tag which uses static access of log() function of Log class

**Parameters**

| | |
|---|---|
| *what* | the message to be logged |
| *log_mode* | the description where the message is outputted ( |

**See also**

Log::MODE::CONSOLE or
Log::MODE::FILE or both)
Log::log()

Definition at line 166 of file Log.hpp.

#### 5.29.1.18 UNDEFINED_COLOR

```
#define UNDEFINED_COLOR(
            x ) ("\033[1;31m" + x + "\033[0m")
```

bold red coloring for console output

Coloring for console output. Unused for file output. See table of codes here: https://en.wikipedia.org/wiki/ANSI_escape_code#graphics

Definition at line 27 of file Log.hpp.

#### 5.29.1.19 WARN

```
#define WARN(
            what ) Log::log(origin(), what, Log::LOG_LEVEL::WARN_LEVEL, meta_type())
```

perform logging of data with the WARN tag

Shortcut macro for logging of data with the WARN tag which uses static access of log() function of Log class

**Parameters**

| | |
|---|---|
| *what* | the message to be logged |
| *log_mode* | the description where the message is outputted ( |

**See also**

Log::MODE::CONSOLE or
Log::MODE::FILE or both)
Log::log()

Definition at line 154 of file Log.hpp.

#### 5.29.1.20 WARN_COLOR

```
#define WARN_COLOR(
            x ) ("\033[33m" + x + "\033[0m")
```

yellow coloring for console output

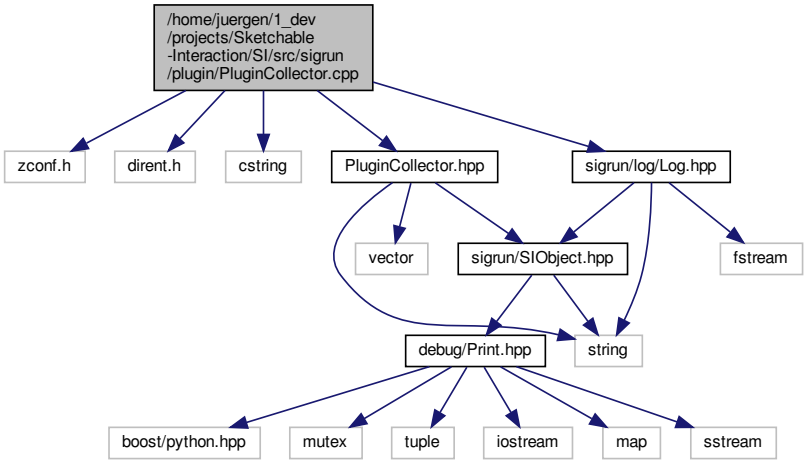
Coloring for console output. Unused for file output. See table of codes here: https://en.wikipedia.org/wiki/ANSI_escape_code#graphics

Definition at line 45 of file Log.hpp.

## 5.30 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/PluginCollector.cpp File Reference

```
#include <zconf.h>
#include <dirent.h>
#include <cstring>
#include "PluginCollector.hpp"
#include "sigrun/log/Log.hpp"
```

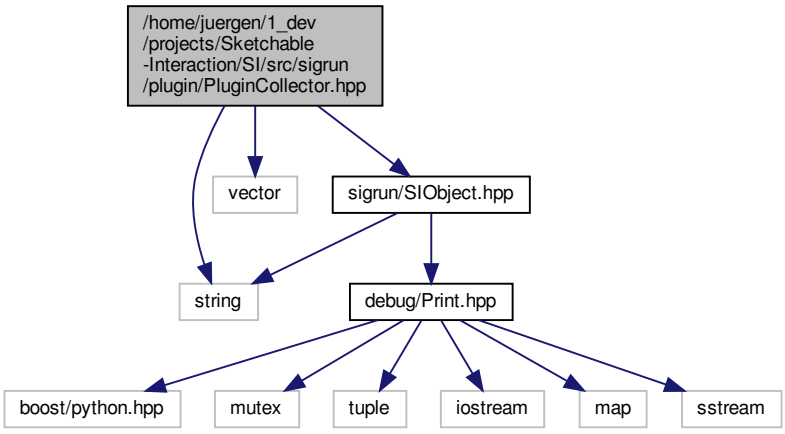
Include dependency graph for PluginCollector.cpp:



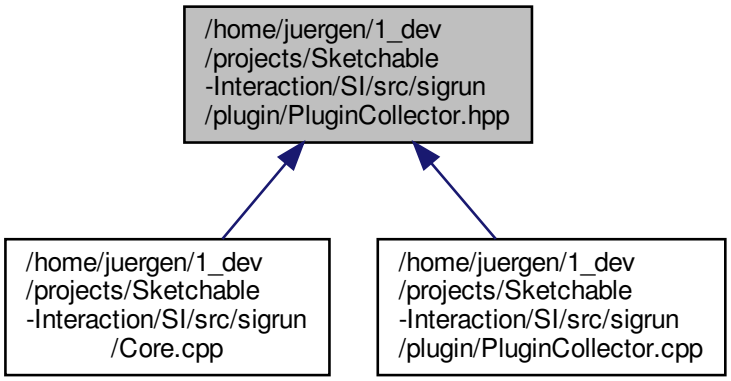

## 5.31 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/PluginCollector.hpp File Reference

```
#include <string>
#include <vector>
#include <sigrun/SIObject.hpp>
```

Include dependency graph for PluginCollector.hpp:




This graph shows which files directly or indirectly include this file:



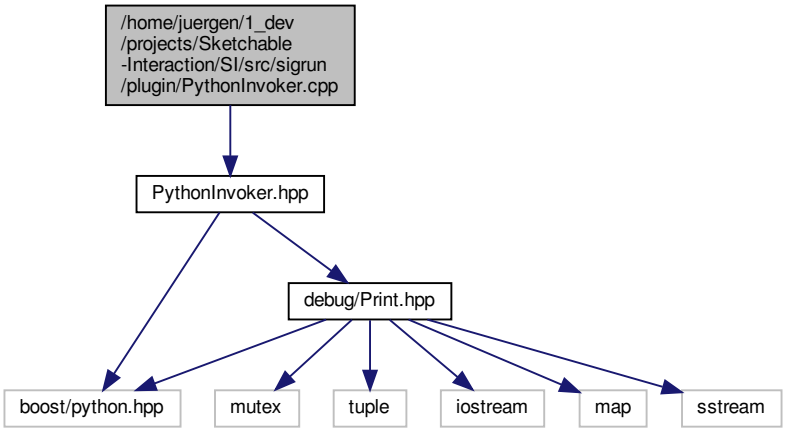

### Classes

- class PluginCollector

## 5.32 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/PythonInvoker.cpp File Reference

```
#include "PythonInvoker.hpp"
```

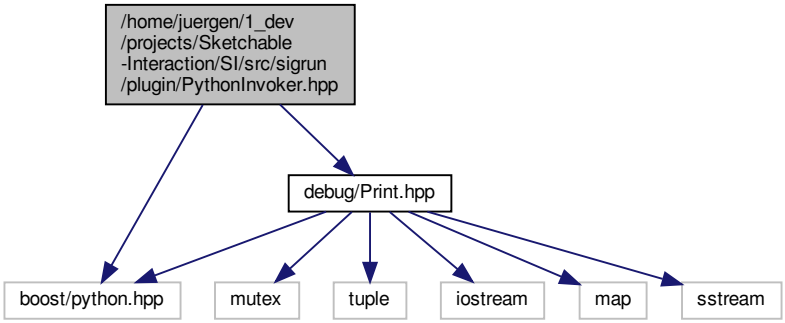
Include dependency graph for PythonInvoker.cpp:



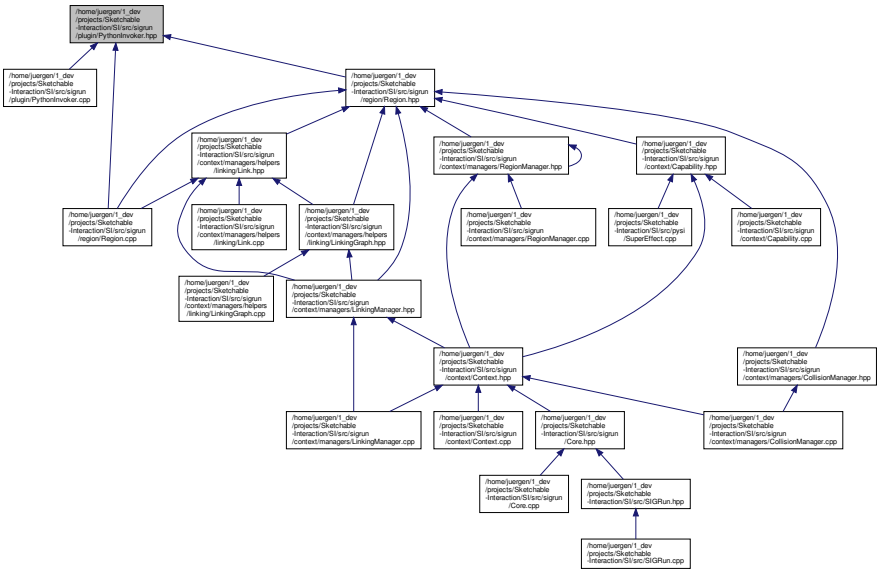

## 5.33 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/PythonInvoker.hpp File Reference

```
#include <boost/python.hpp>
#include "debug/Print.hpp"
```

Include dependency graph for PythonInvoker.hpp:



This graph shows which files directly or indirectly include this file:




## Classes

- class PythonInvoker

## Macros

- #define HANDLE_PYTHON_ERROR

### 5.33.1 Macro Definition Documentation

#### 5.33.1.1 HANDLE_PYTHON_ERROR

```
#define HANDLE_PYTHON_ERROR
```

**Value:**

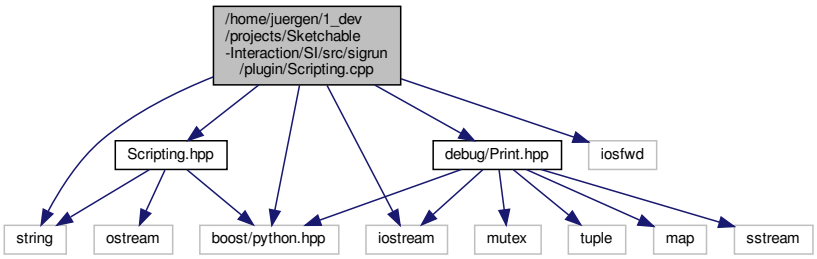
```
catch (const bp::error_already_set&)\
{\
    handle_python_error();\
    return -2;\
}
```

Definition at line 10 of file PythonInvoker.hpp.

## 5.34 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/Scripting.cpp File Reference

```
#include "Scripting.hpp"
#include <iostream>
#include <string>
#include <iosfwd>
#include <boost/python.hpp>
#include <debug/Print.hpp>
```

Include dependency graph for Scripting.cpp:




### Functions

- std::ostream & operator<< (std::ostream &os, const Scripting &scripting)

### 5.34.1 Function Documentation

#### 5.34.1.1 operator<<()

```
std::ostream& operator<< (
            std::ostream & os,
            const Scripting & scripting )
```

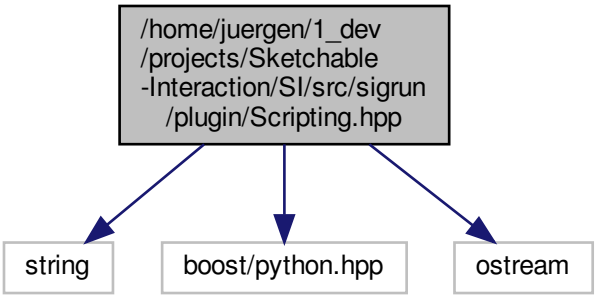
Definition at line 110 of file Scripting.cpp.

## 5.35 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/plugin/Scripting.hpp File Reference

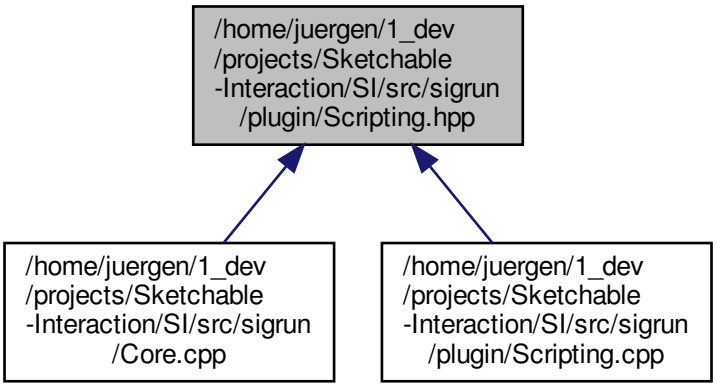
```
#include <string>
#include <boost/python.hpp>
```

```
#include <ostream>
```

Include dependency graph for Scripting.hpp:




This graph shows which files directly or indirectly include this file:




## Classes

- class Scripting

## Functions

- PyObject * PyInit_libPySI (void)

### 5.35.1 Function Documentation

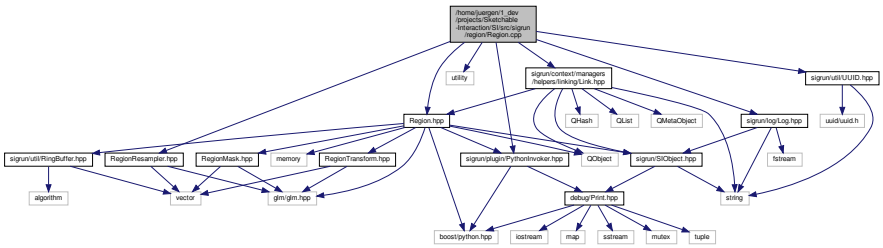
#### 5.35.1.1 PyInit_libPySI()

```
PyObject* PyInit_libPySI (
            void  )
```

## 5.36 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/Region.cpp File Reference

```
#include "sigrun/plugin/PythonInvoker.hpp"
#include "Region.hpp"
#include <utility>
#include "RegionResampler.hpp"
#include <sigrun/log/Log.hpp>
#include <sigrun/context/managers/helpers/linking/Link.hpp>
#include <sigrun/util/UUID.hpp>
```

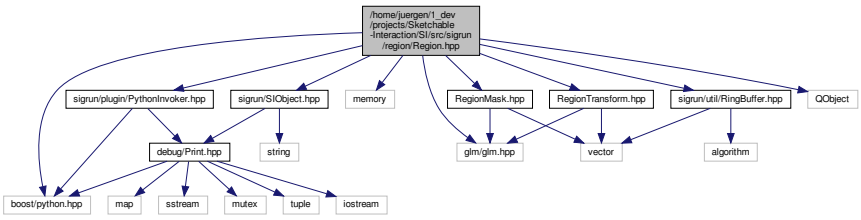
Include dependency graph for Region.cpp:



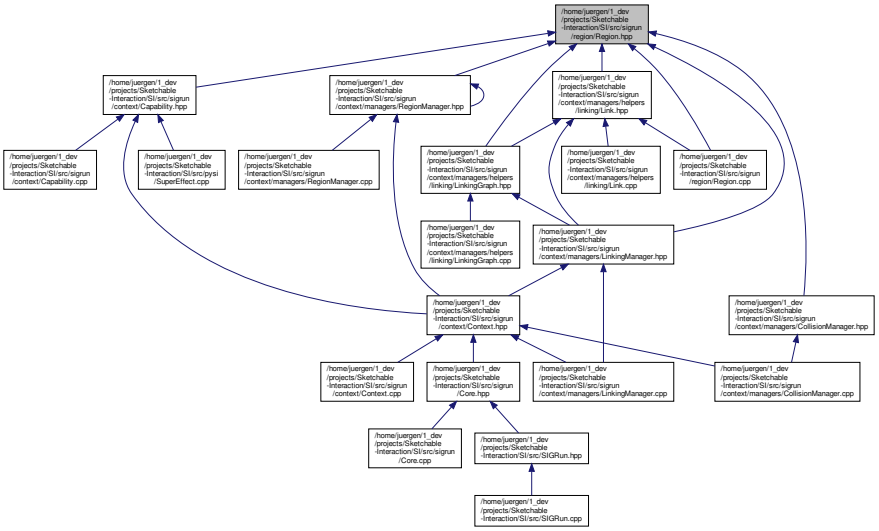

## 5.37 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/Region.hpp File Reference

```
#include <boost/python.hpp>
#include <memory>
#include <glm/glm.hpp>
#include "RegionMask.hpp"
#include "RegionTransform.hpp"
#include "sigrun/plugin/PythonInvoker.hpp"
#include <sigrun/SIObject.hpp>
#include <sigrun/util/RingBuffer.hpp>
#include <QObject>
```

Include dependency graph for Region.hpp:



This graph shows which files directly or indirectly include this file:



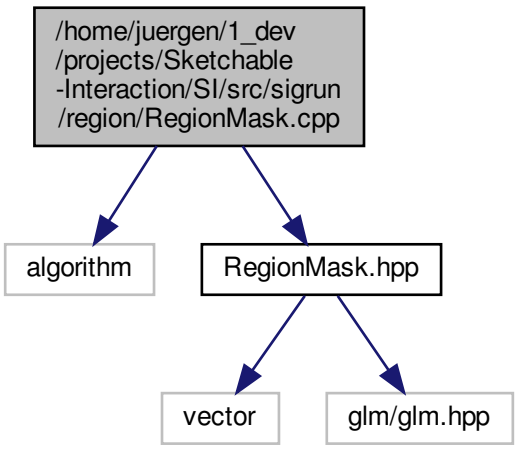

### Classes

- class Region

## 5.38 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/Region↩ Mask.cpp File Reference

```
#include <algorithm>
#include "RegionMask.hpp"
```

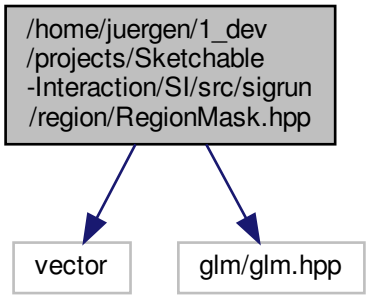
Include dependency graph for RegionMask.cpp:



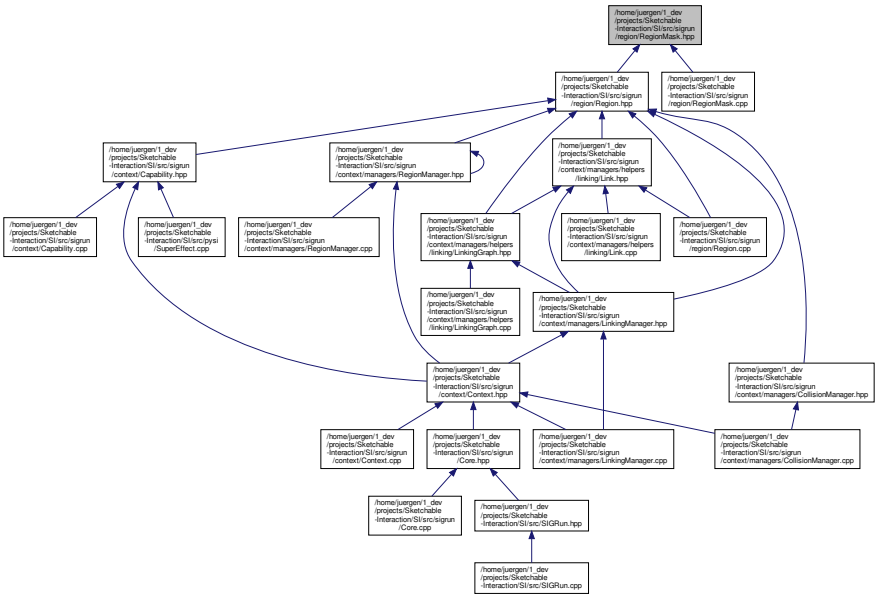
## 5.39 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionMask.hpp File Reference

```
#include <vector>
#include <glm/glm.hpp>
```

Include dependency graph for RegionMask.hpp:




This graph shows which files directly or indirectly include this file:



### Classes

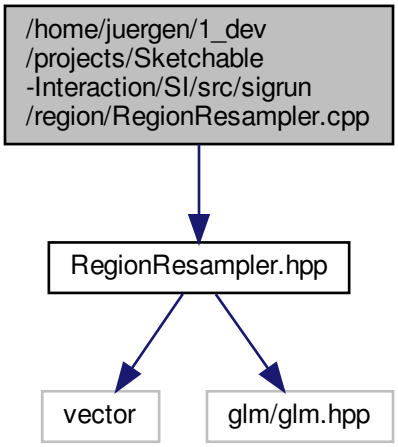
- class RegionMask

  *RegionMask class which stores a bit array used for true collision testing.*

## 5.40 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionResampler.cpp File Reference

```
#include "RegionResampler.hpp"
```

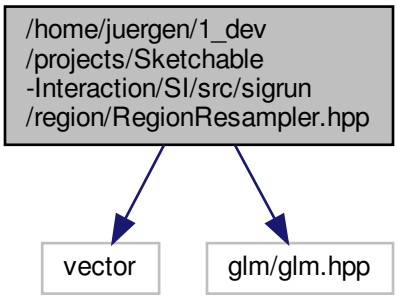
Include dependency graph for RegionResampler.cpp:



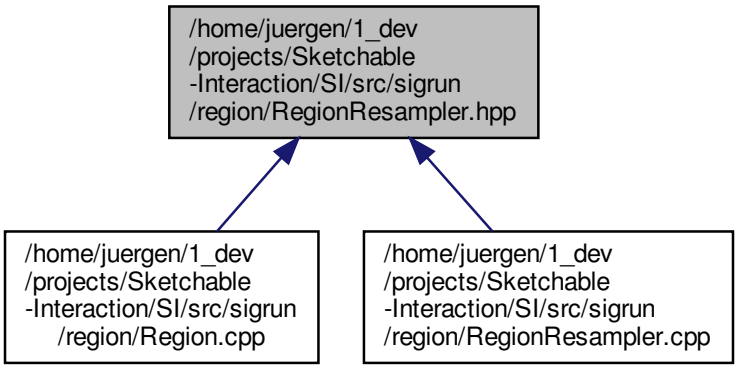

## 5.41 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionResampler.hpp File Reference

```
#include <vector>
#include <glm/glm.hpp>
```

Include dependency graph for RegionResampler.hpp:



This graph shows which files directly or indirectly include this file:



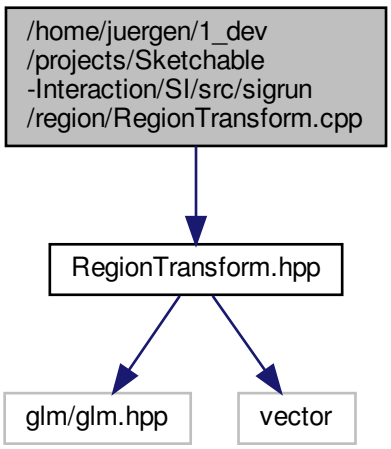

### Classes

- class RegionResampler

## 5.42 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionTransform.cpp File Reference

```
#include "RegionTransform.hpp"
```

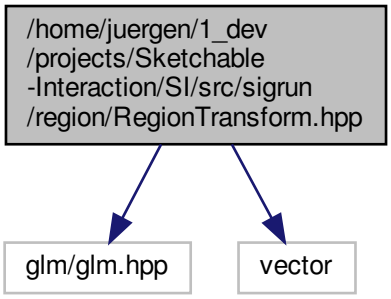
Include dependency graph for RegionTransform.cpp:



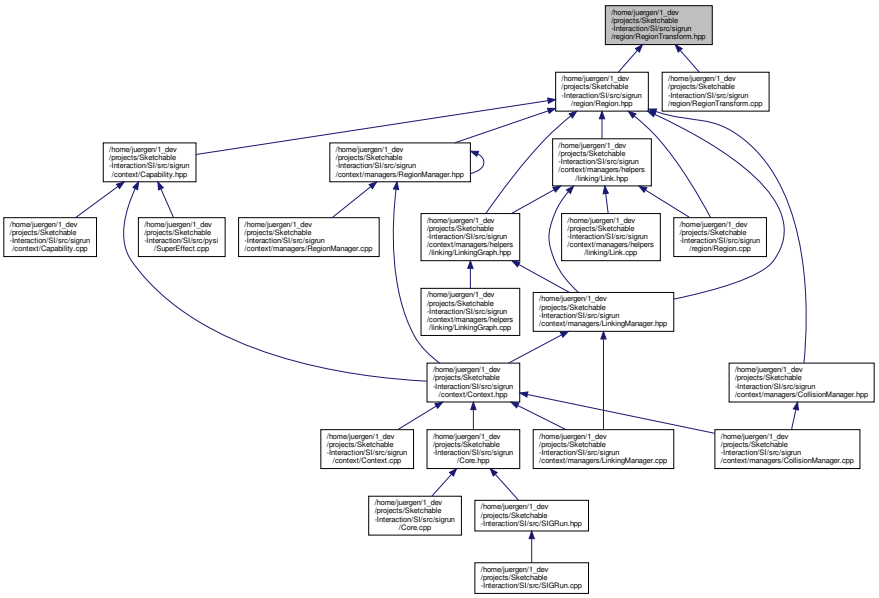
## 5.43 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/region/RegionTransform.hpp File Reference

```
#include <glm/glm.hpp>
#include <vector>
```

Include dependency graph for RegionTransform.hpp:




This graph shows which files directly or indirectly include this file:



### Classes

- class RegionTransform

  *RegionTransform class storing the relative translation, rotation and scale of a contour.*

### Macros

- #define PI_DIV_180 (float) 0.0174532925199
  *quivalent to M_PI / 180.0*

### 5.43.1 Macro Definition Documentation

#### 5.43.1.1 PI_DIV_180

```
#define PI_DIV_180 (float) 0.0174532925199
```

quivalent to M_PI / 180.0

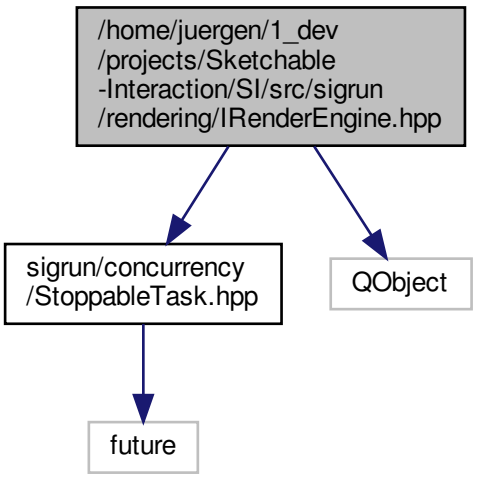
Equivalent to M_PI / 180.0. Can be used to convert angles given in degrees to radians.

Definition at line 15 of file RegionTransform.hpp.

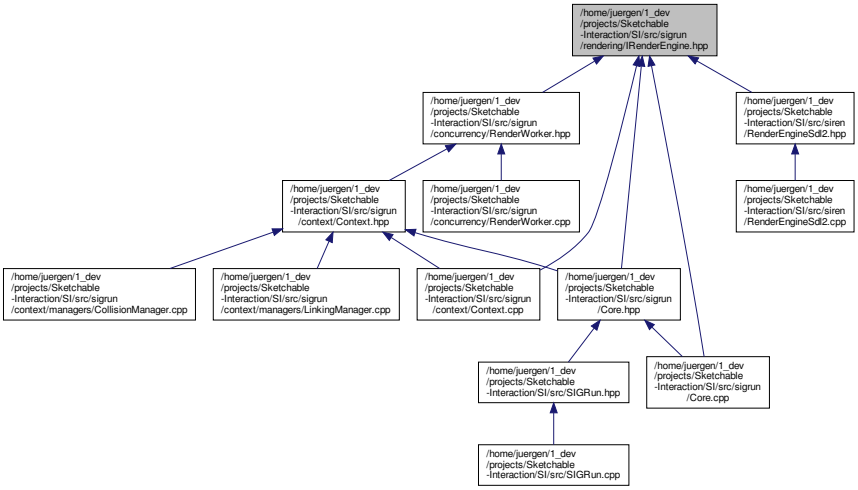
## 5.44 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/rendering/IRenderEngine.hpp File Reference

```
#include <sigrun/concurrency/StoppableTask.hpp>
#include <QObject>
```

Include dependency graph for IRenderEngine.hpp:



This graph shows which files directly or indirectly include this file:



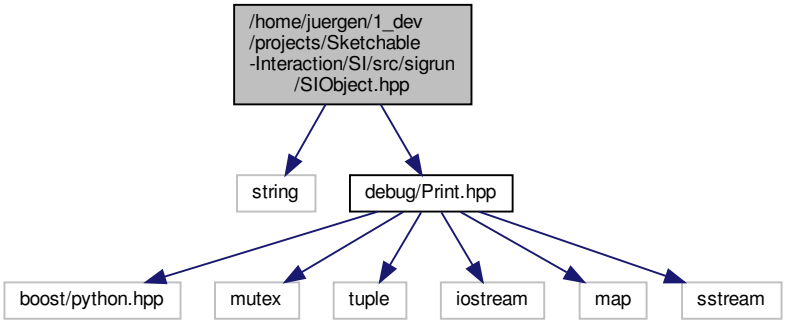

### Classes

- class IRenderEngine

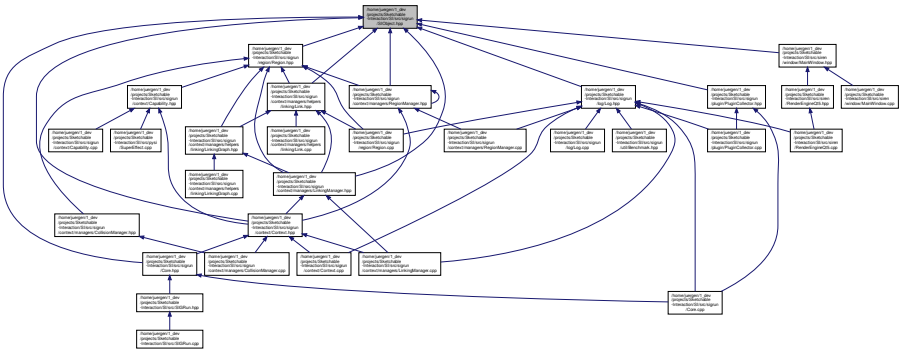
## 5.45 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/SIObject.hpp File Reference

```
#include <string>
#include <debug/Print.hpp>
```

Include dependency graph for SIObject.hpp:



This graph shows which files directly or indirectly include this file:



## Classes

- class SIObject

  *A meta class from which other classes are derived from to register them as SIObject meta types.*

## Macros

- #define __CLASS_NAME__

  *macro for extracting the name of the class to be registered as an SIObject*
- #define SIOBJECT(origin)

  *macro for registering another class as SIObject*
- #define SIGRUN SIOBJECT(SIGRUN);
- #define SIREN SIOBJECT(SIREN);

### 5.45.1 Macro Definition Documentation

#### 5.45.1.1 __CLASS_NAME__

```
#define __CLASS_NAME__
```

**Value:**

```
(\
{\
    std::string s = std::string(__PRETTY_FUNCTION__).substr(0, strchr(__PRETTY_FUNCTION__, ':') -
       __PRETTY_FUNCTION__);\
    std::transform(s.begin(), s.end(), s.begin(), ::toupper);\
    s;\
})
```

macro for extracting the name of the class to be registered as an SIObject

The macro computes the name of a class which is to be registered as an SIObject. After retrieving the name of the class, the macro makes transforms the result string to uppercase.

**Returns**

a std::string containing the uppercase name of the class to be registered as an SIObject

Definition at line 16 of file SIObject.hpp.

#### 5.45.1.2 SIGRUN

```
#define SIGRUN SIOBJECT(SIGRUN);
```

Definition at line 41 of file SIObject.hpp.

#### 5.45.1.3 SIOBJECT

```
#define SIOBJECT(
               origin )
```

**Value:**

```
(\
{\
    d_meta_type = __CLASS_NAME__;\
    d_origin = #origin;\
});
```

macro for registering another class as SIObject

The macro is a shortcut for registering other classes which are derived from SIObject as such a SIObject. Syntax: class A: public SIObject {SIOBJECT ... };

Definition at line 35 of file SIObject.hpp.

#### 5.45.1.4 SIREN

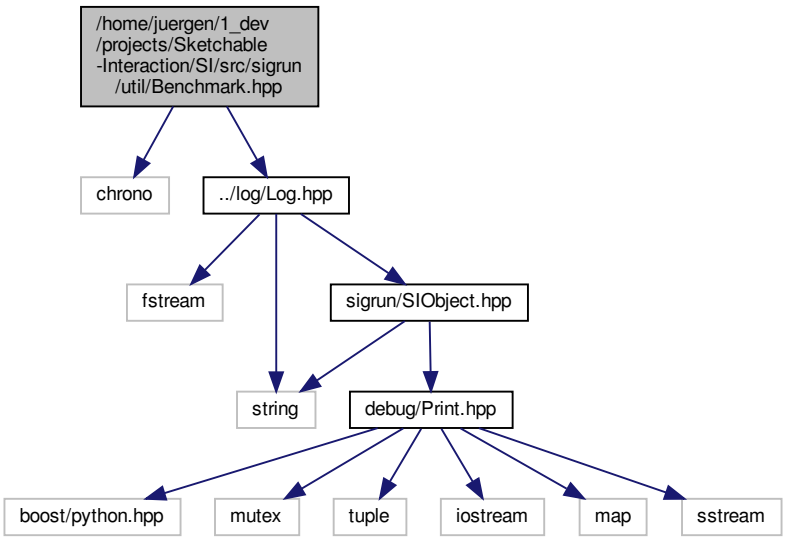
```
#define SIREN SIOBJECT(SIREN);
```

Definition at line 42 of file SIObject.hpp.

## 5.46 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/util/Benchmark.hpp File Reference

```
#include <chrono>
#include "../log/Log.hpp"
```

Include dependency graph for Benchmark.hpp:




## Classes

- class BenchmarkTimer

## Macros

- #define SI_BENCHMARK(...) { SI::BenchmarkTimer timer; __VA_ARGS__ }

### 5.46.1 Macro Definition Documentation

#### 5.46.1.1 SI_BENCHMARK

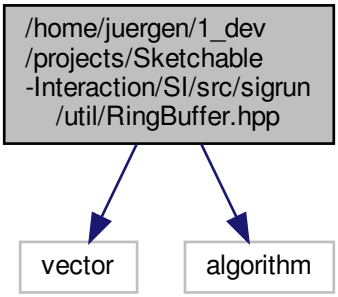
```
#define SI_BENCHMARK(
              ...  ) { SI::BenchmarkTimer timer; __VA_ARGS__ }
```

Definition at line 9 of file Benchmark.hpp.

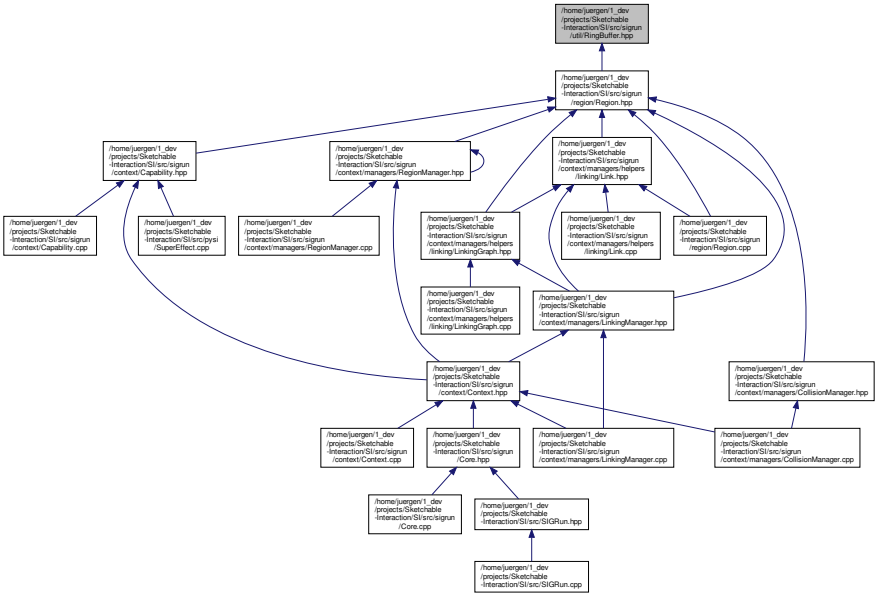
## 5.47 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/util/RingBuffer.hpp File Reference

```
#include <vector>
#include <algorithm>
```

Include dependency graph for RingBuffer.hpp:




This graph shows which files directly or indirectly include this file:



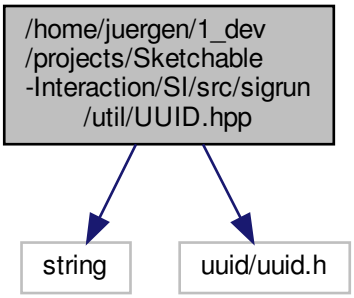
### Classes

- class RingBuffer< T >

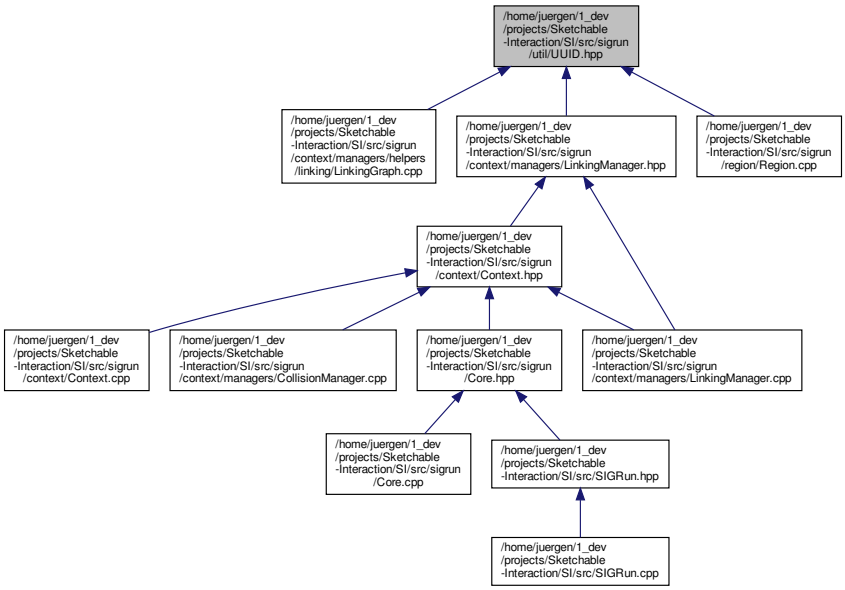
## 5.48 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/sigrun/util/UUID.hpp File Reference

```
#include <string>
#include <uuid/uuid.h>
```

Include dependency graph for UUID.hpp:




This graph shows which files directly or indirectly include this file:



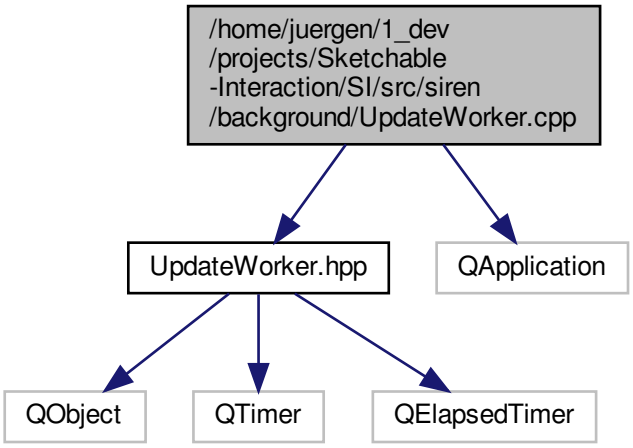

### Classes

- class UUID

## 5.49 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/background/UpdateWorker.cpp File Reference

```
#include "UpdateWorker.hpp"
#include <QApplication>
```

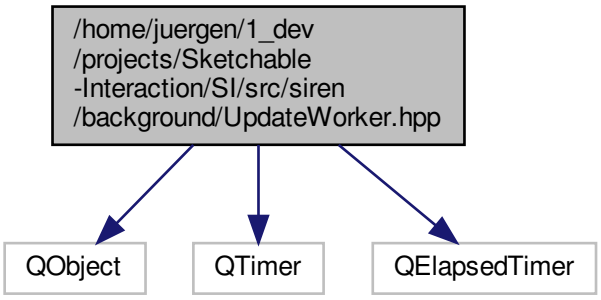
Include dependency graph for UpdateWorker.cpp:



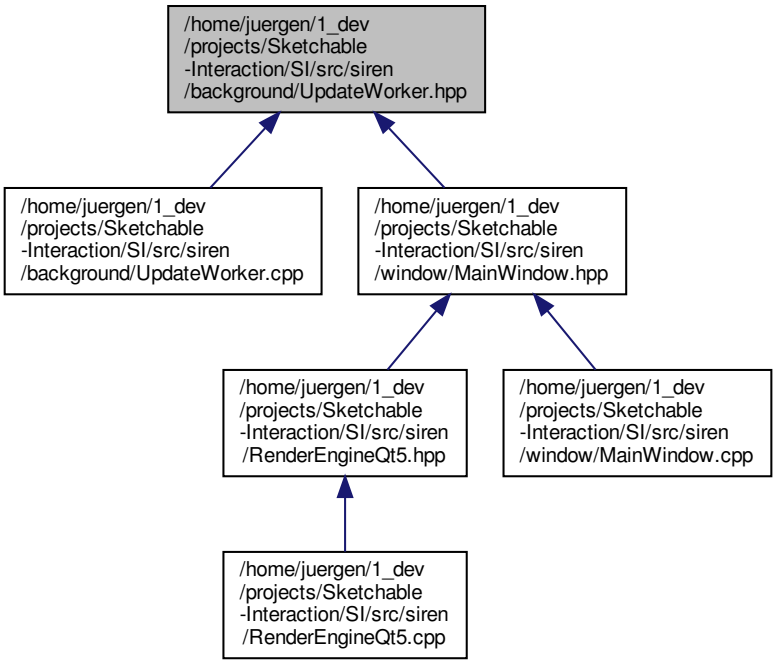

## 5.50 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/background/UpdateWorker.hpp File Reference

```
#include <QObject>
#include <QTimer>
#include <QElapsedTimer>
```

Include dependency graph for UpdateWorker.hpp:



This graph shows which files directly or indirectly include this file:



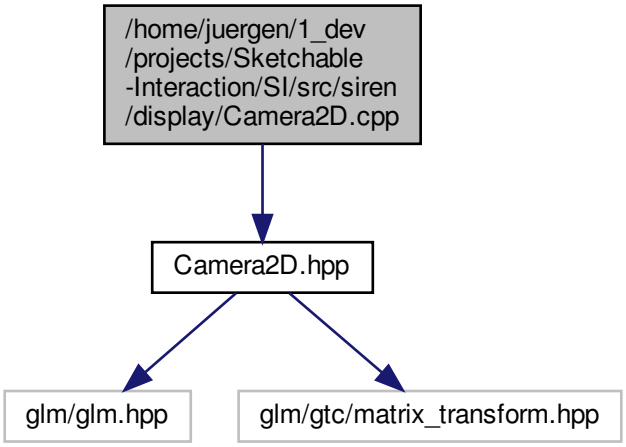

### Classes

- class UpdateWorker

## 5.51 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Camera2D.cpp File Reference

```
#include "Camera2D.hpp"
```

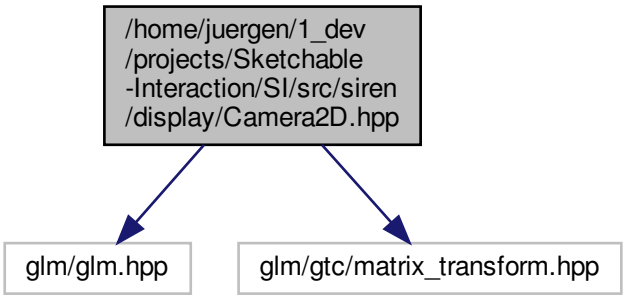
Include dependency graph for Camera2D.cpp:



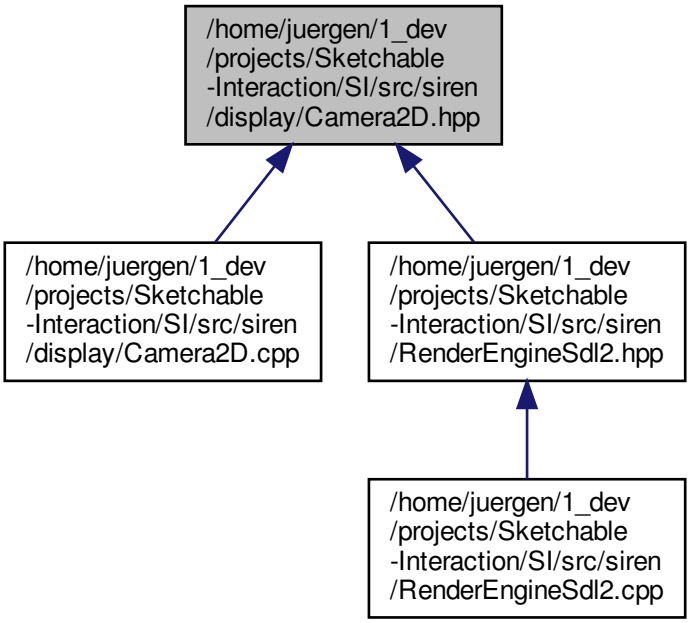

## 5.52 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Camera2D.hpp File Reference

```
#include <glm/glm.hpp>
#include <glm/gtc/matrix_transform.hpp>
```

Include dependency graph for Camera2D.hpp:



This graph shows which files directly or indirectly include this file:



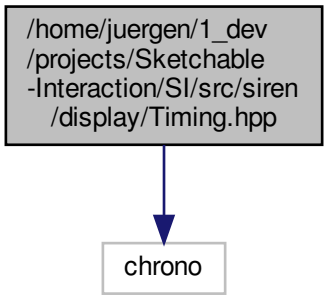

### Classes

- class Camera2D

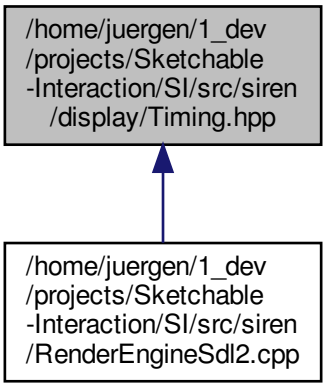
## 5.53 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Timing.hpp File Reference

```
#include <chrono>
```

Include dependency graph for Timing.hpp:



This graph shows which files directly or indirectly include this file:



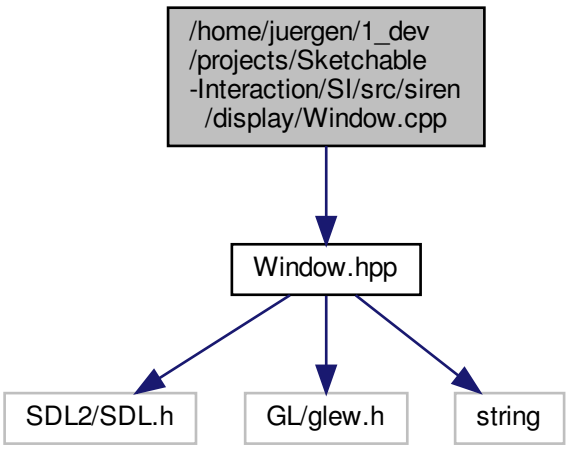

### Classes

- class Time

## 5.54 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Window.cpp File Reference

```
#include "Window.hpp"
```

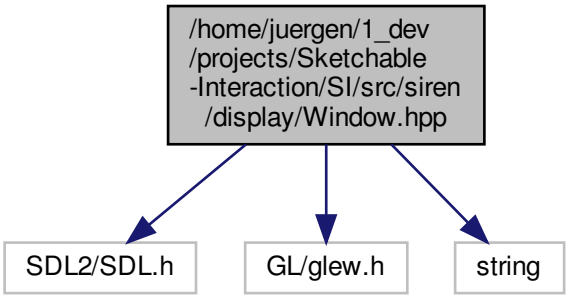
Include dependency graph for Window.cpp:



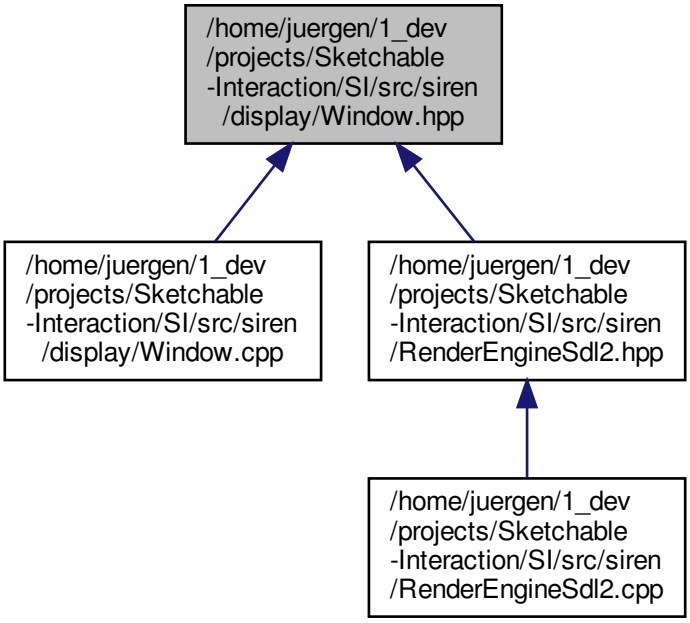
## 5.55 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/display/Window.hpp File Reference

```
#include <SDL2/SDL.h>
#include <GL/glew.h>
#include <string>
```

Include dependency graph for Window.hpp:




This graph shows which files directly or indirectly include this file:



### Classes

- class Window

### Enumerations

- enum WindowFlags { INVISIBLE = 0x1, FULLSCREEN = 0x2, BORDERLESS = 0x4 }

### 5.55.1 Enumeration Type Documentation

#### 5.55.1.1 WindowFlags

```
enum WindowFlags
```

**Enumerator**

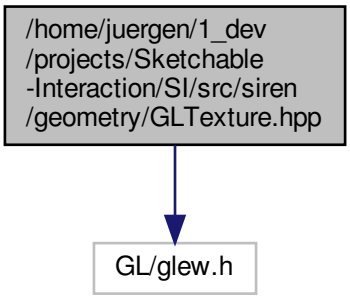
| | |
|---|---|
| INVISIBLE | |
| FULLSCREEN | |
| BORDERLESS | |

Definition at line 11 of file Window.hpp.

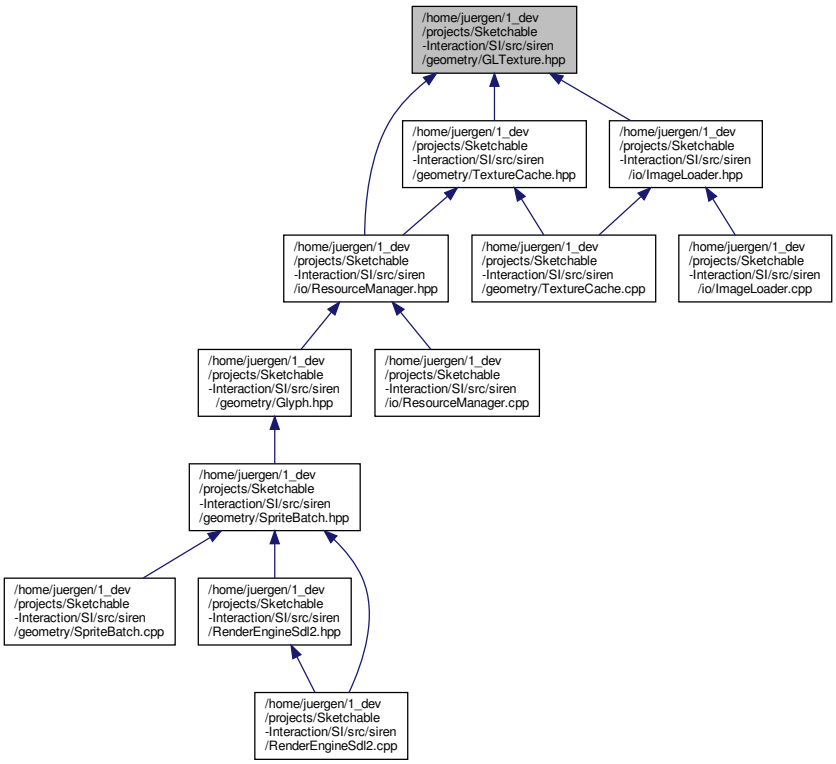
## 5.56 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/GL↩ Texture.hpp File Reference

```
#include <GL/glew.h>
```

Include dependency graph for GLTexture.hpp:



This graph shows which files directly or indirectly include this file:



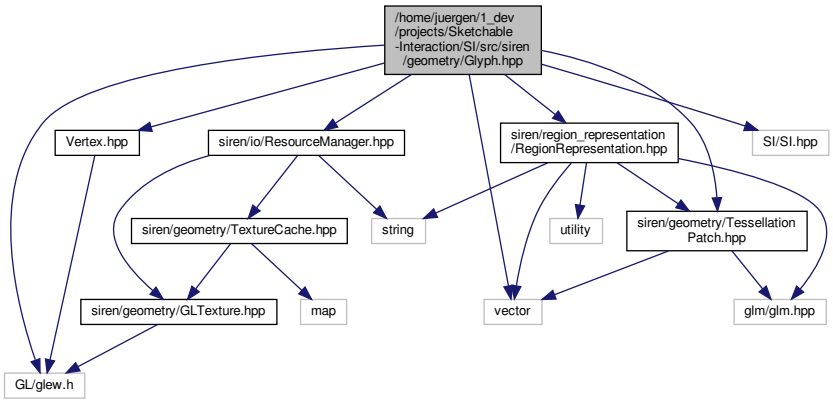

### Classes

- struct GLTexture

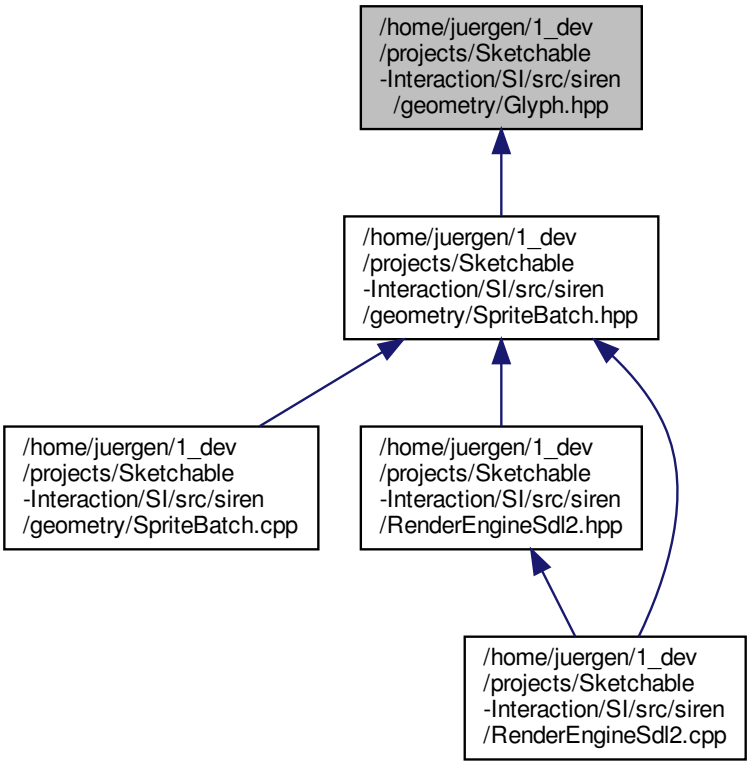
## 5.57 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Glyph.hpp File Reference

```
#include <GL/glew.h>
#include "Vertex.hpp"
#include <vector>
#include <SI/SI.hpp>
#include <siren/io/ResourceManager.hpp>
#include <siren/geometry/TessellationPatch.hpp>
#include <siren/region_representation/RegionRepresentation.hpp>
```

Include dependency graph for Glyph.hpp:




This graph shows which files directly or indirectly include this file:




## Classes

- class Glyph

## Enumerations

- enum GlyphSortType { GlyphSortType::NONE, GlyphSortType::FRONT_TO_BACK, GlyphSortType::BACK_TO_FRONT, GlyphSortType::TEXTURE }

### 5.57.1 Enumeration Type Documentation

#### 5.57.1.1 GlyphSortType

```
enum GlyphSortType  [strong]
```

**Enumerator**

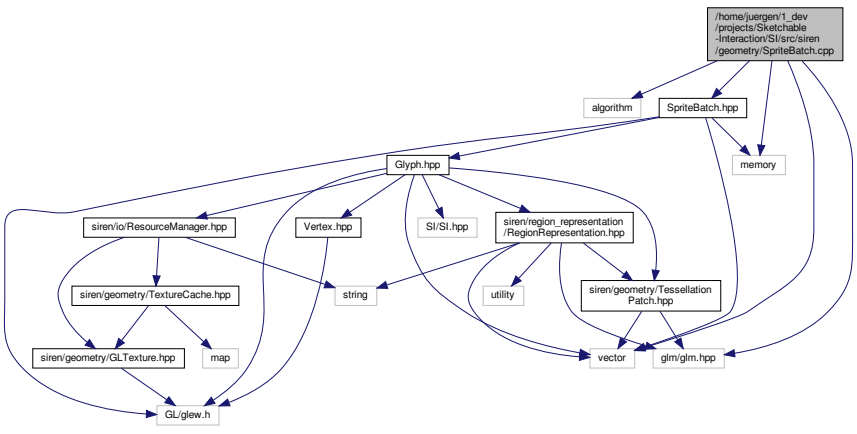
| | |
|---|---|
| NONE | |
| FRONT_TO_BACK | |
| BACK_TO_FRONT | |
| TEXTURE | |

Definition at line 13 of file Glyph.hpp.

## 5.58 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/SpriteBatch.cpp File Reference

```
#include <algorithm>
#include "SpriteBatch.hpp"
#include <vector>
#include <memory>
#include <glm/glm.hpp>
```

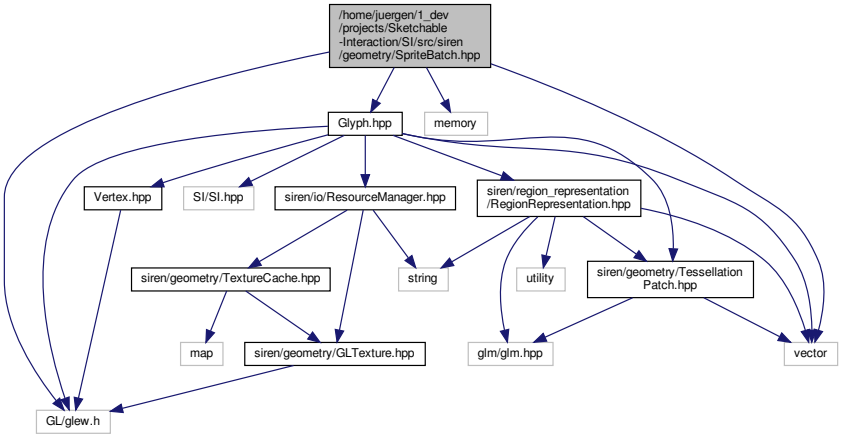
Include dependency graph for SpriteBatch.cpp:



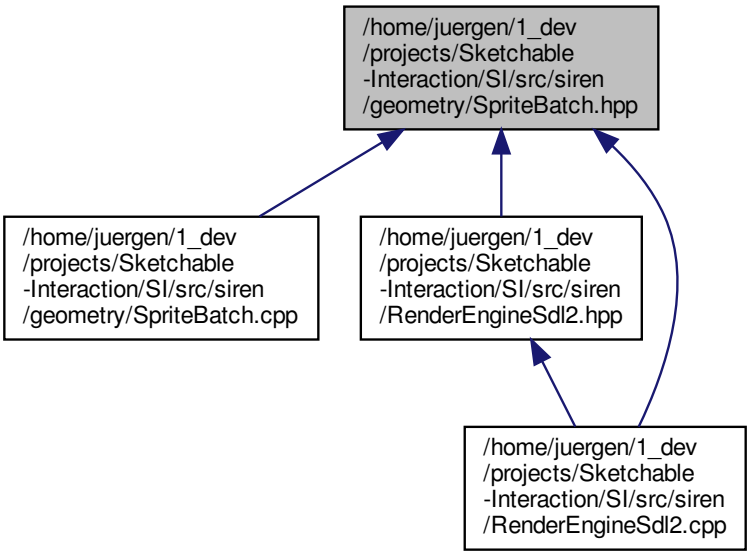
## 5.59 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/SpriteBatch.hpp File Reference

```
#include <GL/glew.h>
#include <vector>
#include <memory>
#include "Glyph.hpp"
```

Include dependency graph for SpriteBatch.hpp:




This graph shows which files directly or indirectly include this file:



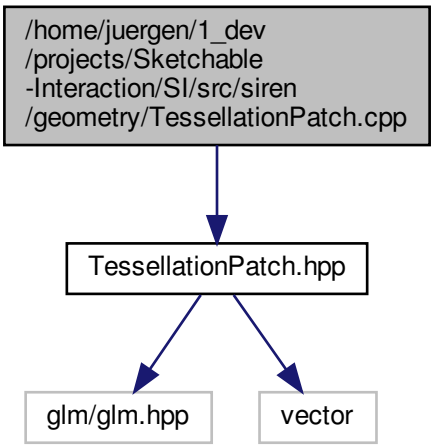
### Classes

- class RenderBatch
- class SpriteBatch

## 5.60 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/TessellationPatch.cpp File Reference

```
#include "TessellationPatch.hpp"
```

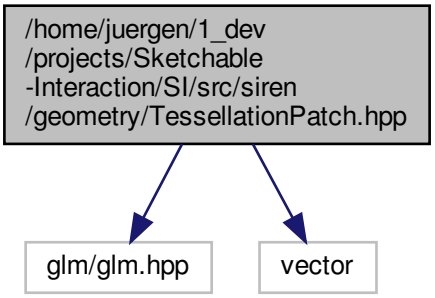
Include dependency graph for TessellationPatch.cpp:



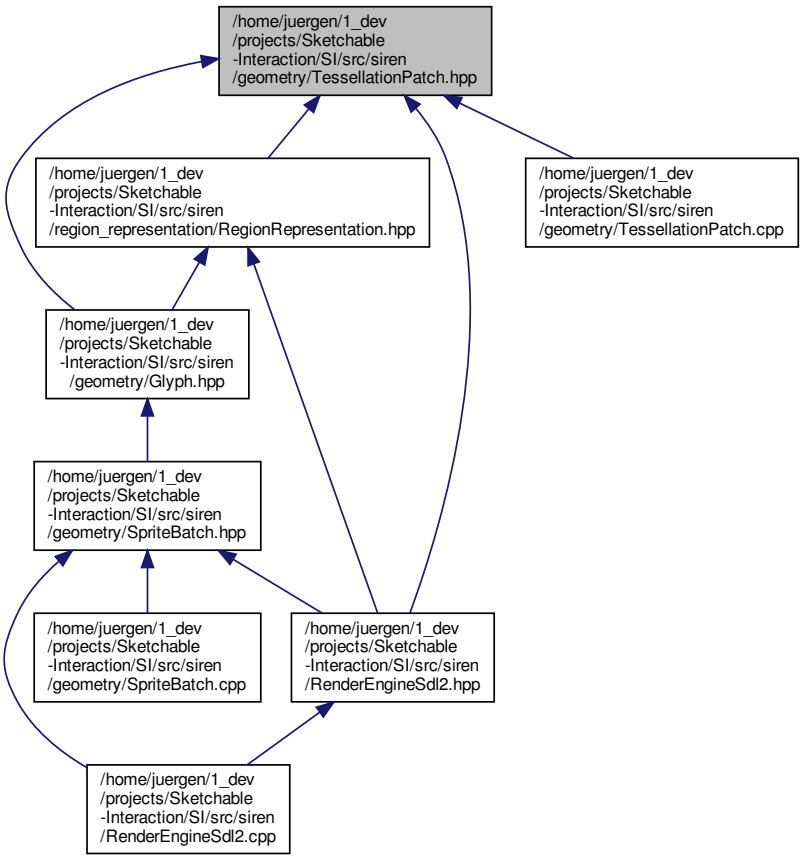

## 5.61 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/TessellationPatch.hpp File Reference

```
#include <glm/glm.hpp>
#include <vector>
```

Include dependency graph for TessellationPatch.hpp:




This graph shows which files directly or indirectly include this file:



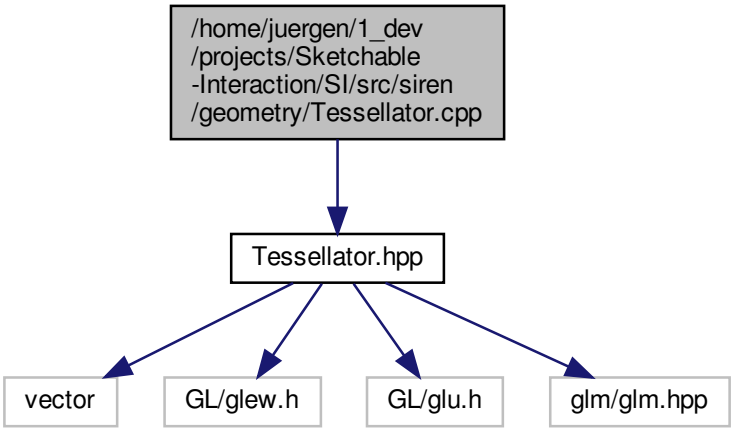

## Classes

- class TesselationPatch

## 5.62 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Tessellator.cpp File Reference

```
#include "Tessellator.hpp"
```

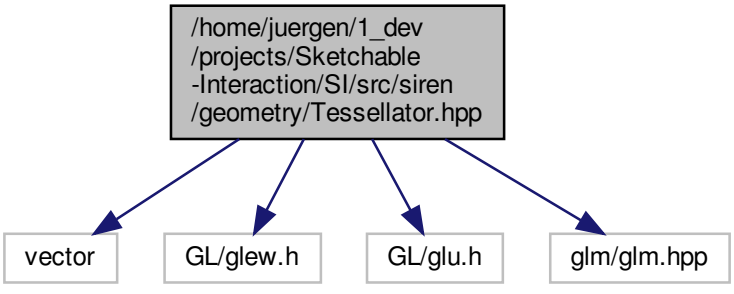
Include dependency graph for Tessellator.cpp:



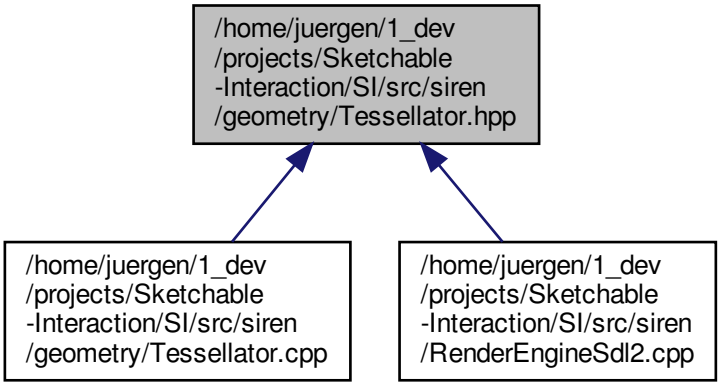

## 5.63 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Tessellator.hpp File Reference

```
#include <vector>
#include <GL/glew.h>
#include <GL/glu.h>
#include <glm/glm.hpp>
```

Include dependency graph for Tessellator.hpp:



This graph shows which files directly or indirectly include this file:




## Classes

- class Tessellator

## Macros

- #define TESSELATION_CALLBACK

### 5.63.1 Macro Definition Documentation

#### 5.63.1.1 TESSELATION_CALLBACK

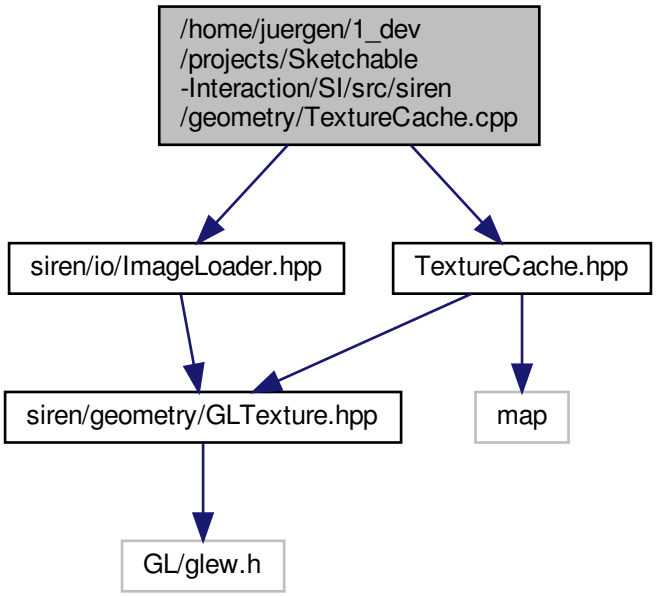
```
#define TESSELATION_CALLBACK
```

Definition at line 12 of file Tessellator.hpp.

## 5.64 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/TextureCache.cpp File Reference

```
#include <siren/io/ImageLoader.hpp>
#include "TextureCache.hpp"
```

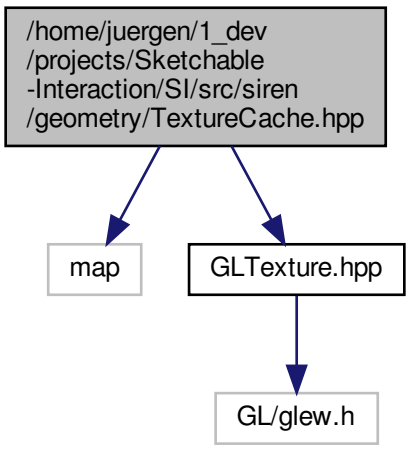
Include dependency graph for TextureCache.cpp:



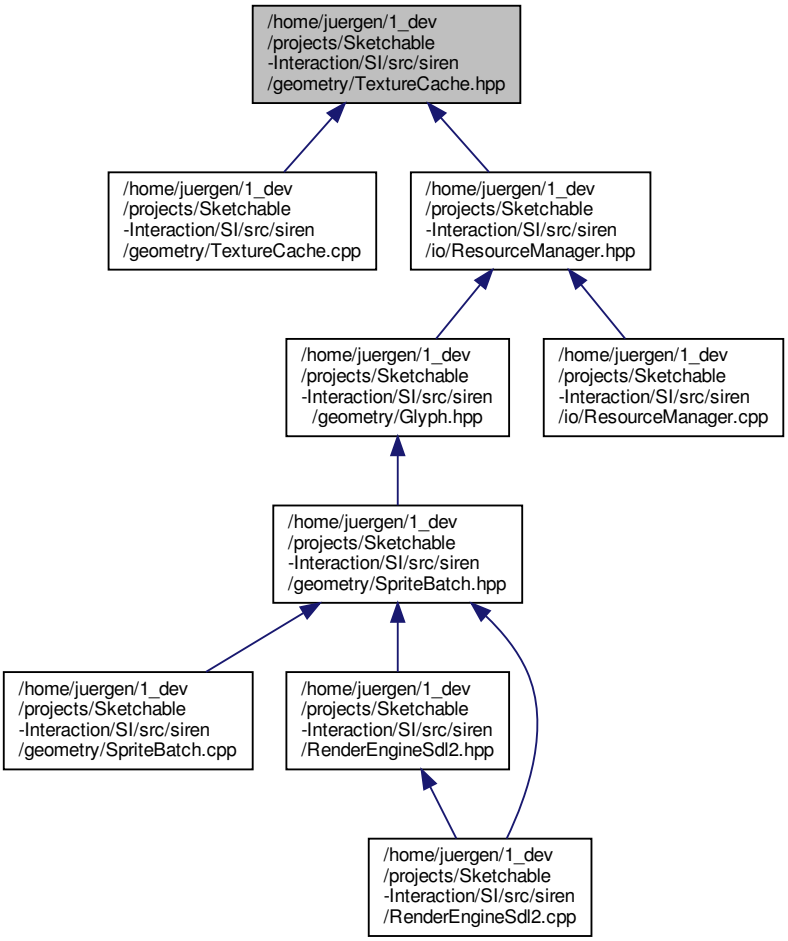

## 5.65 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/TextureCache.hpp File Reference

```
#include <map>
#include "GLTexture.hpp"
```

Include dependency graph for TextureCache.hpp:



This graph shows which files directly or indirectly include this file:



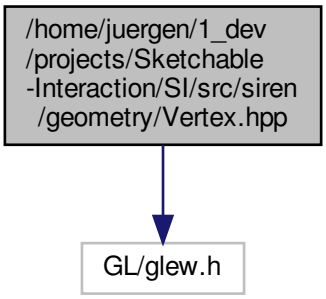

### Classes

- class TextureCache

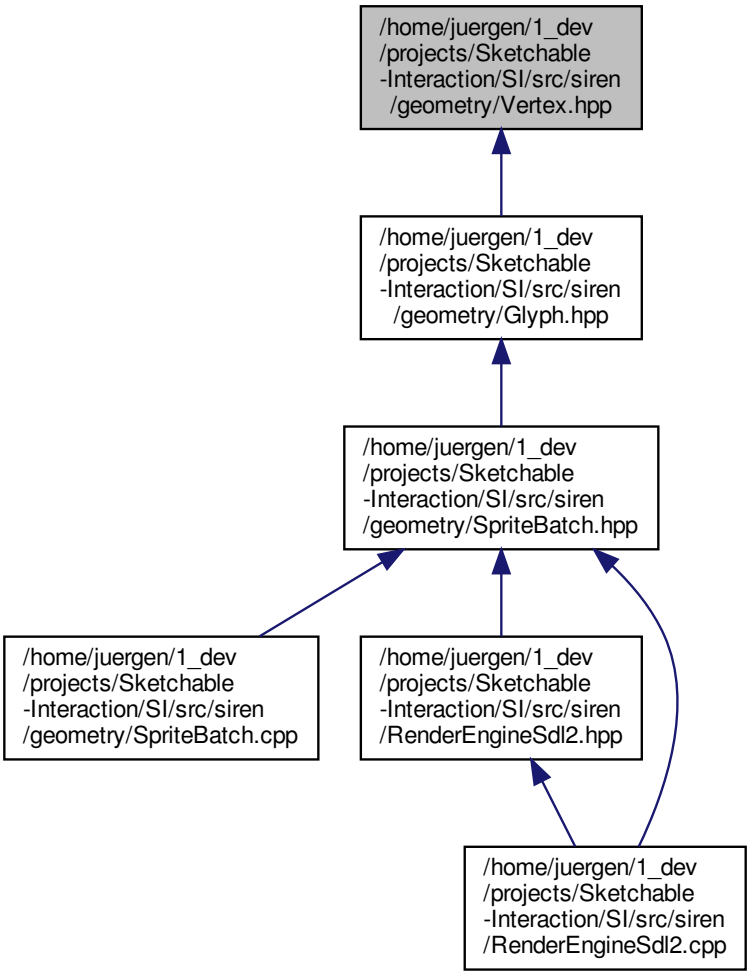
## 5.66 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/geometry/Vertex.hpp File Reference

```
#include <GL/glew.h>
```

Include dependency graph for Vertex.hpp:



This graph shows which files directly or indirectly include this file:




### Classes

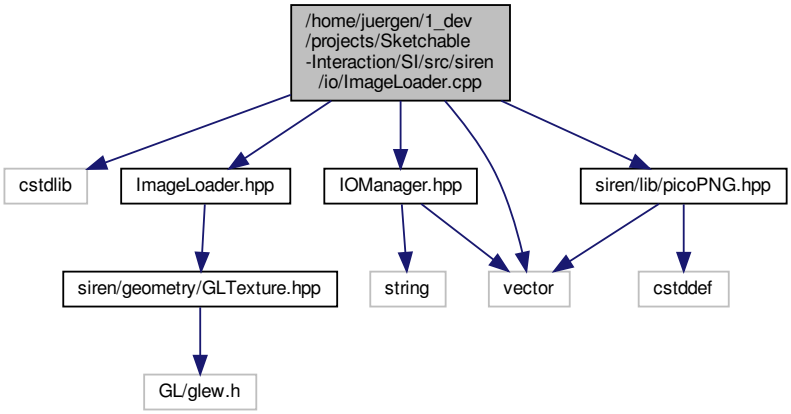
- struct Position
- struct Color
- struct UV
- struct Vertex

## 5.67 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/ImageLoader.cpp File Reference

```
#include <cstdlib>
#include "ImageLoader.hpp"
#include "IOManager.hpp"
```

```
#include <vector>
#include <siren/lib/picoPNG.hpp>
```

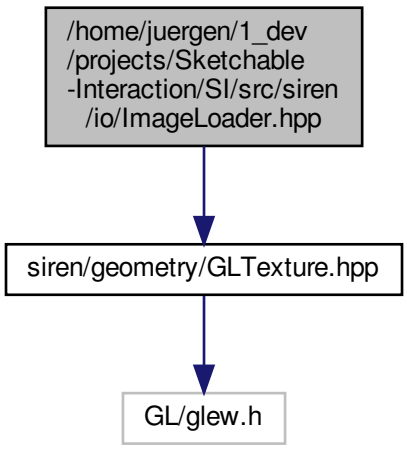
Include dependency graph for ImageLoader.cpp:



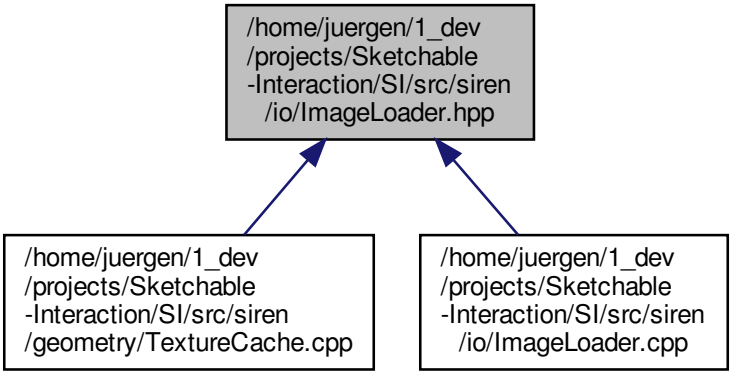

## 5.68 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/ImageLoader.hpp File Reference

```
#include <siren/geometry/GLTexture.hpp>
```

Include dependency graph for ImageLoader.hpp:



This graph shows which files directly or indirectly include this file:



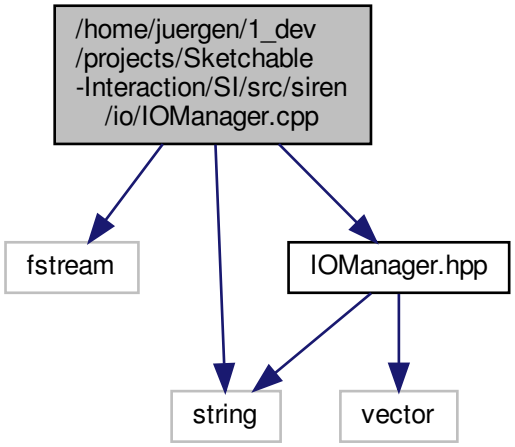

### Classes

- class ImageLoader

## 5.69 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/IOManager.cpp File Reference

```
#include <fstream>
#include <string>
#include "IOManager.hpp"
```

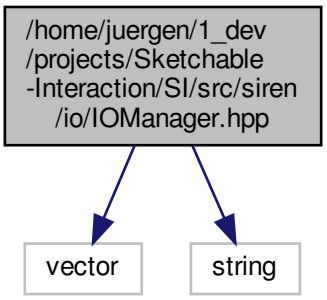
Include dependency graph for IOManager.cpp:



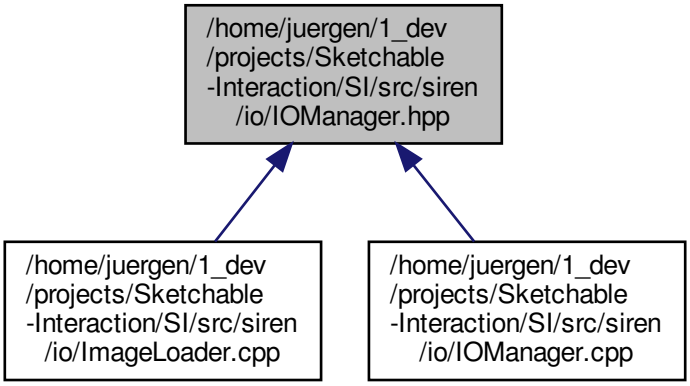
## 5.70 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/IOManager.hpp File Reference

```
#include <vector>
#include <string>
```

Include dependency graph for IOManager.hpp:




This graph shows which files directly or indirectly include this file:



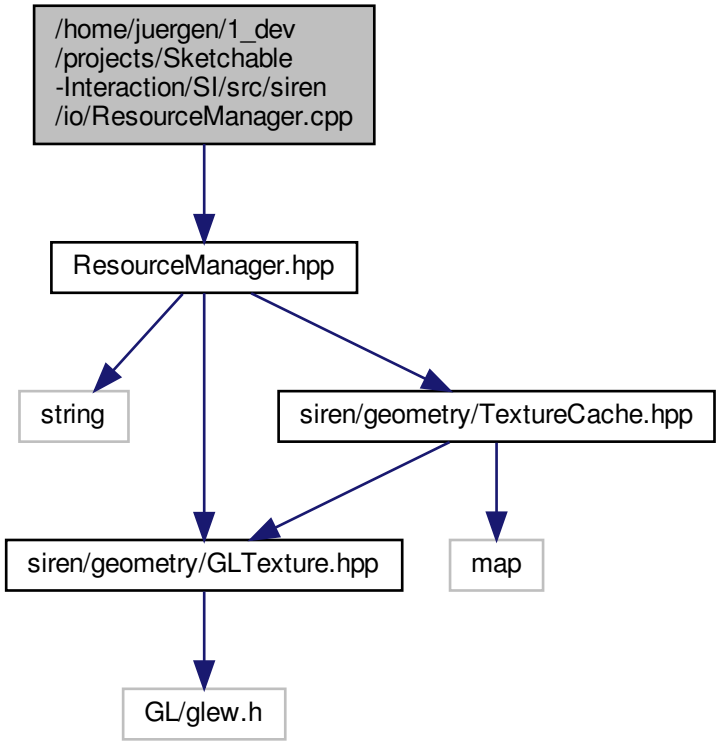

### Classes

- class IOManager

## 5.71 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/ResourceManager.cpp File Reference

```
#include "ResourceManager.hpp"
```

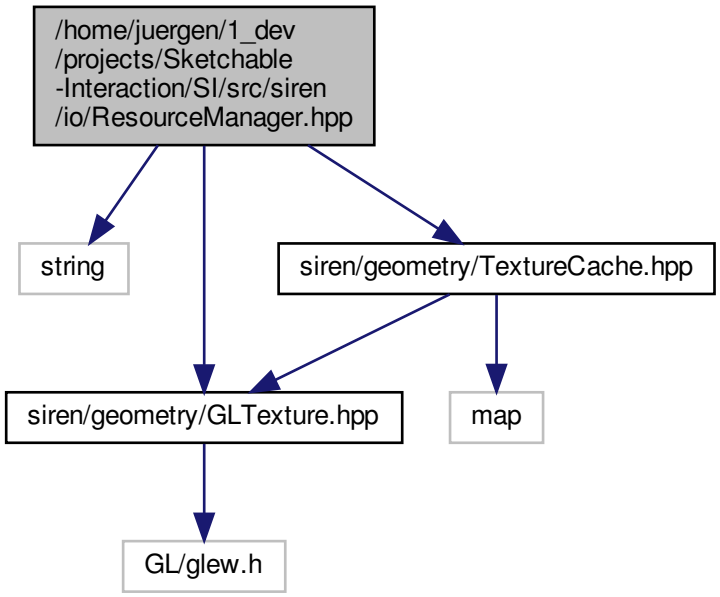
Include dependency graph for ResourceManager.cpp:



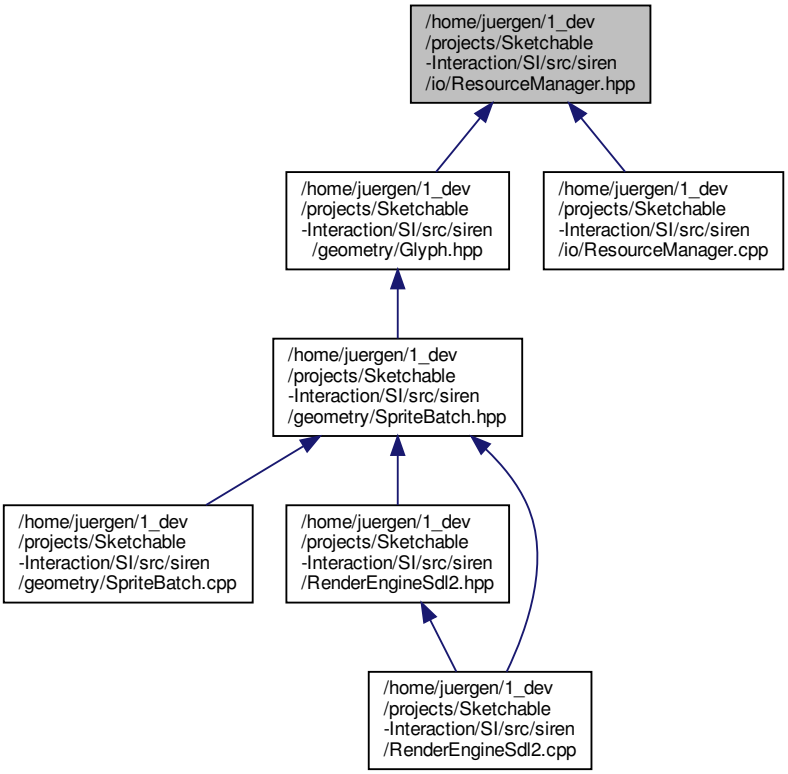

## 5.72 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/io/ResourceManager.hpp File Reference

```
#include <string>
#include <siren/geometry/GLTexture.hpp>
#include <siren/geometry/TextureCache.hpp>
```

Include dependency graph for ResourceManager.hpp:



This graph shows which files directly or indirectly include this file:



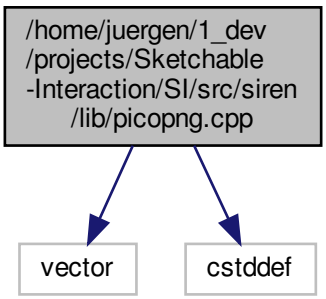

### Classes

- class ResourceManager

## 5.73 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/lib/picopng.cpp File Reference

```
#include <vector>
#include <cstddef>
```

Include dependency graph for picopng.cpp:




## Functions

- int decodePNG (std::vector< unsigned char > &out_image, unsigned long &image_width, unsigned long &image_height, const unsigned char *in_png, size_t in_size, bool convert_to_rgba32)

### 5.73.1 Function Documentation

#### 5.73.1.1 decodePNG()

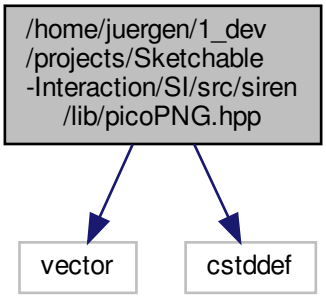
```
int decodePNG (
             std::vector< unsigned char > & out_image,
             unsigned long & image_width,
             unsigned long & image_height,
             const unsigned char * in_png,
             size_t in_size,
             bool convert_to_rgba32 )
```

Definition at line 25 of file picopng.cpp.

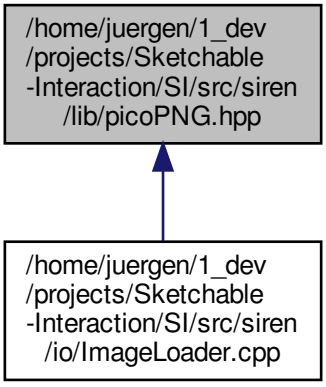
## 5.74 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/lib/picoPNG.hpp File Reference

```
#include <vector>
#include <cstddef>
```

Include dependency graph for picoPNG.hpp:




This graph shows which files directly or indirectly include this file:




## Functions

- int decodePNG (std::vector< unsigned char > &out_image, unsigned long &image_width, unsigned long &image_height, const unsigned char *in_png, size_t in_size, bool convert_to_rgba32=true)

### 5.74.1 Function Documentation

#### 5.74.1.1 decodePNG()

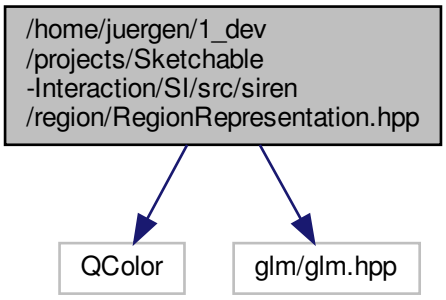
```
int decodePNG (
             std::vector< unsigned char > & out_image,
             unsigned long & image_width,
             unsigned long & image_height,
             const unsigned char * in_png,
             size_t in_size,
             bool convert_to_rgba32 = true )
```

Definition at line 25 of file picopng.cpp.

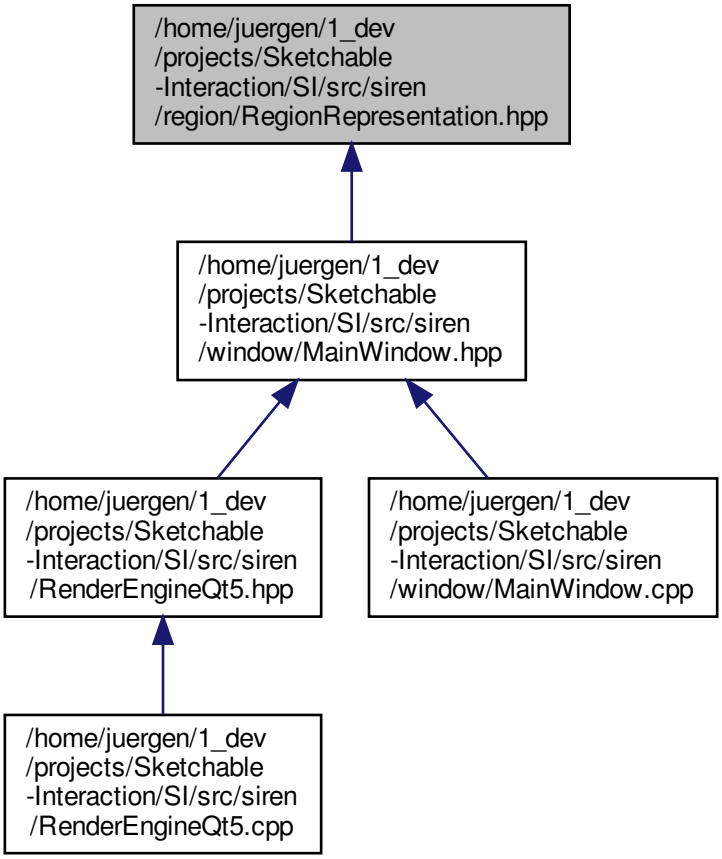
## 5.75 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/region/RegionRepresentation.hpp File Reference

```
#include <QColor>
#include <glm/glm.hpp>
```

Include dependency graph for RegionRepresentation.hpp:



This graph shows which files directly or indirectly include this file:




### Classes

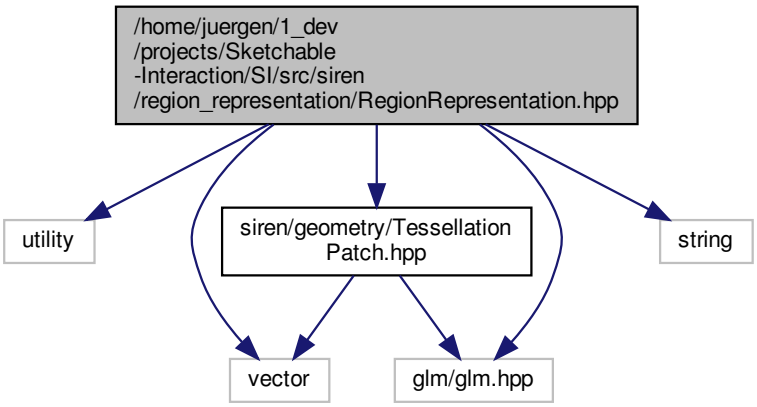
- struct RegionRepresentation

## 5.76 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/region_representation/↩ RegionRepresentation.hpp File Reference

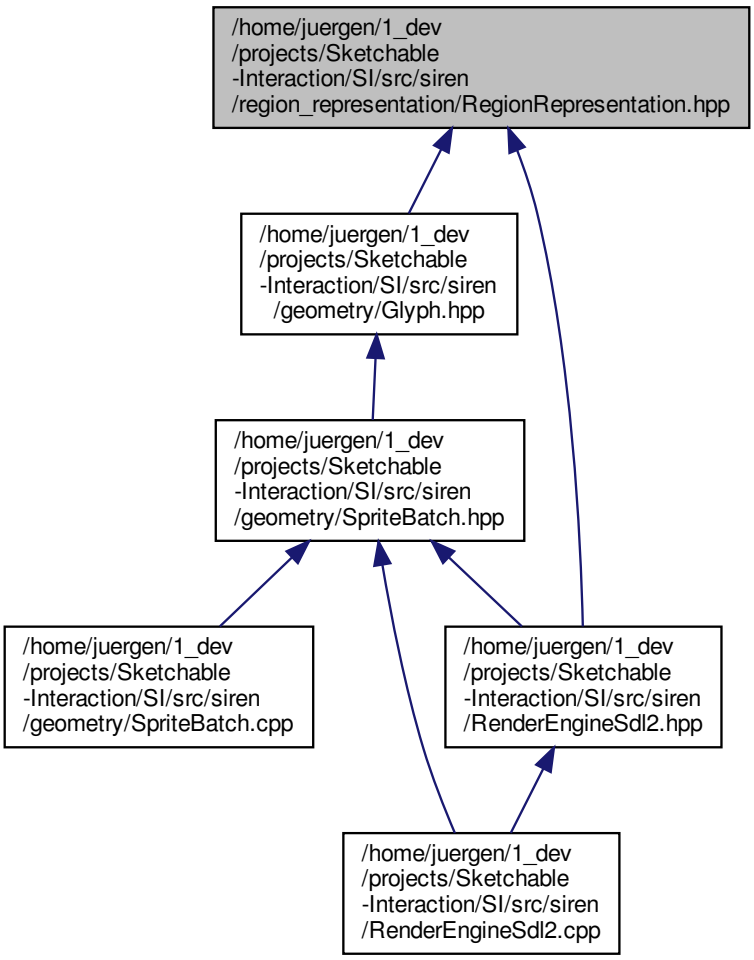
```
#include <utility>
#include <vector>
#include <glm/glm.hpp>
#include <siren/geometry/TessellationPatch.hpp>
```

```
#include <string>
```

Include dependency graph for RegionRepresentation.hpp:



This graph shows which files directly or indirectly include this file:



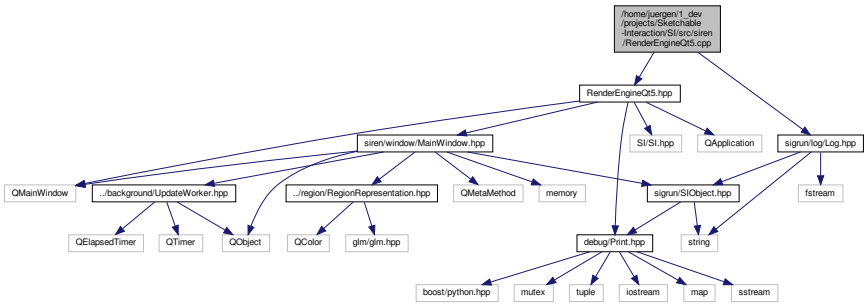

### Classes

- struct RegionRepresentation

## 5.77 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/RenderEngine↩ Qt5.cpp File Reference

```
#include "RenderEngineQt5.hpp"
#include <sigrun/log/Log.hpp>
```

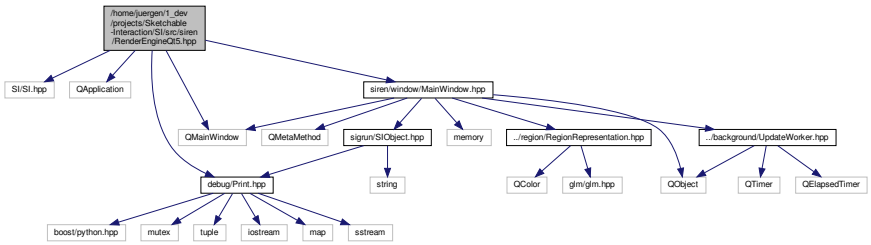
Include dependency graph for RenderEngineQt5.cpp:



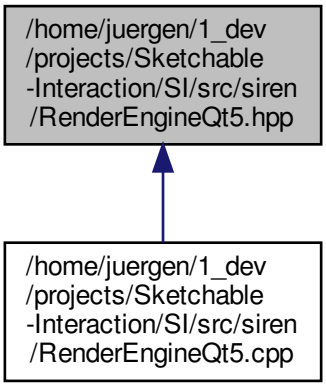

## 5.78 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/RenderEngine↩ Qt5.hpp File Reference

```
#include <SI/SI.hpp>
#include <QApplication>
#include <QMainWindow>
#include <debug/Print.hpp>
#include <siren/window/MainWindow.hpp>
```

Include dependency graph for RenderEngineQt5.hpp:



This graph shows which files directly or indirectly include this file:



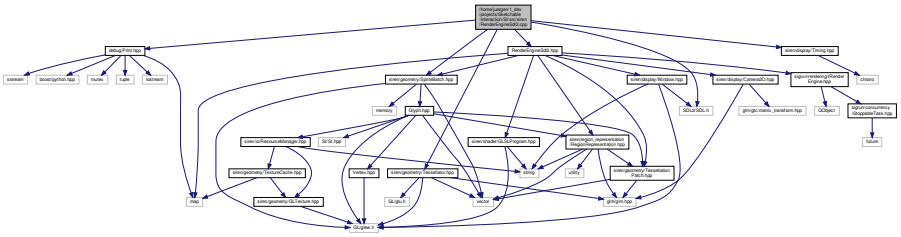

### Classes

- class RenderEngineQT5

## 5.79 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/RenderEngine↩ Sdl2.cpp File Reference

```
#include <debug/Print.hpp>
#include "RenderEngineSdl2.hpp"
#include <SDL2/SDL.h>
#include <siren/geometry/SpriteBatch.hpp>
#include <siren/geometry/Tessellator.hpp>
#include <siren/display/Timing.hpp>
```

Include dependency graph for RenderEngineSdl2.cpp:

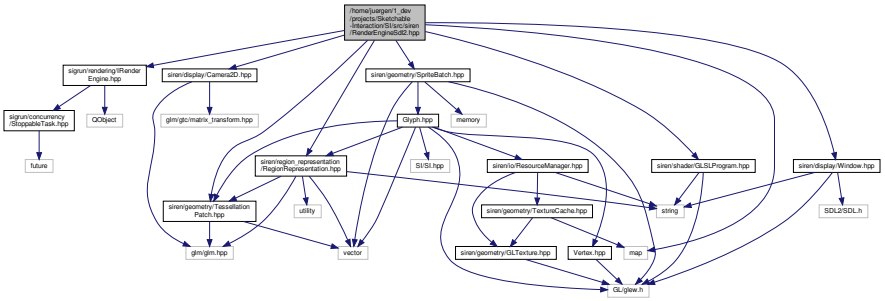


## 5.80 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/RenderEngine↩ Sdl2.hpp File Reference

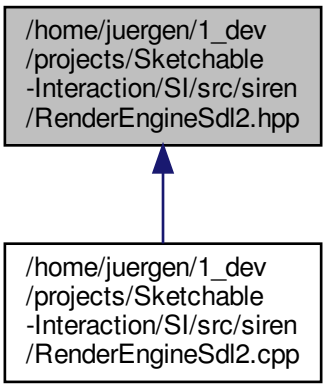
```
#include <sigrun/rendering/IRenderEngine.hpp>
#include <siren/display/Camera2D.hpp>
```

```
#include <siren/shader/GLSLProgram.hpp>
#include <siren/geometry/SpriteBatch.hpp>
#include <siren/display/Window.hpp>
#include <siren/geometry/TessellationPatch.hpp>
#include <map>
#include <siren/region_representation/RegionRepresentation.hpp>
```

Include dependency graph for RenderEngineSdl2.hpp:




This graph shows which files directly or indirectly include this file:




## Classes

- class RenderEngineSDL2

## Typedefs

- typedef std::unique_ptr< Camera2D > Camera2D_ptr
- typedef std::unique_ptr< GLSLProgram > GLSLProgram_ptr
- typedef std::unique_ptr< SpriteBatch > SpriteBatch_ptr

## Enumerations

- enum STATE { STATE::ON, STATE::OFF }

### 5.80.1 Typedef Documentation

#### 5.80.1.1 Camera2D_ptr

```
typedef std::unique_ptr<Camera2D> Camera2D_ptr
```

Definition at line 24 of file RenderEngineSdl2.hpp.

#### 5.80.1.2 GLSLProgram_ptr

```
typedef std::unique_ptr<GLSLProgram> GLSLProgram_ptr
```

Definition at line 25 of file RenderEngineSdl2.hpp.

#### 5.80.1.3 SpriteBatch_ptr

```
typedef std::unique_ptr<SpriteBatch> SpriteBatch_ptr
```

Definition at line 26 of file RenderEngineSdl2.hpp.

### 5.80.2 Enumeration Type Documentation

#### 5.80.2.1 STATE

```
enum STATE  [strong]
```

**Enumerator**

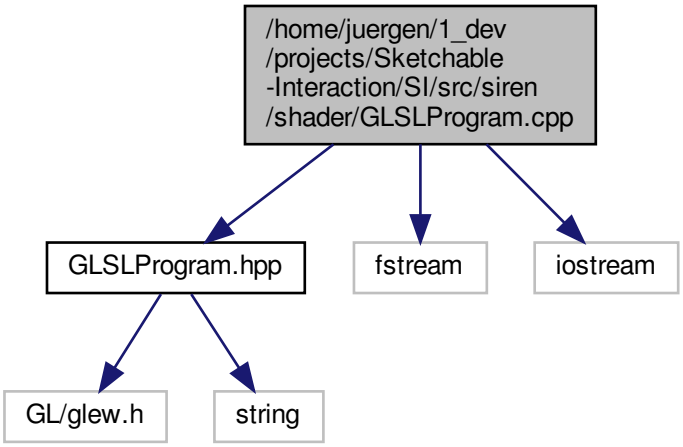
| | |
|---|---|
| ON | |
| OFF | |

Definition at line 18 of file RenderEngineSdl2.hpp.

## 5.81 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/shader/GLSLProgram.cpp File Reference

```
#include "GLSLProgram.hpp"
#include <fstream>
#include <iostream>
```

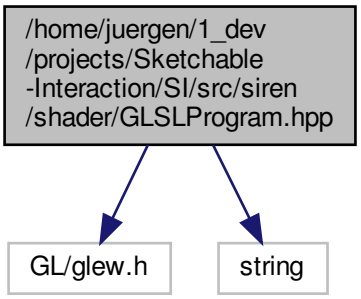
Include dependency graph for GLSLProgram.cpp:



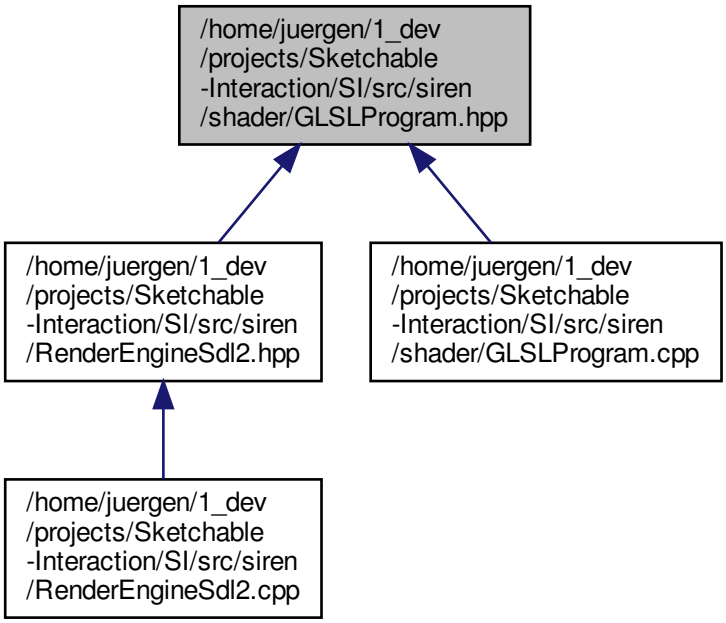

## 5.82 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/shader/GLSLProgram.hpp File Reference

```
#include <GL/glew.h>
#include <string>
```

Include dependency graph for GLSLProgram.hpp:



This graph shows which files directly or indirectly include this file:



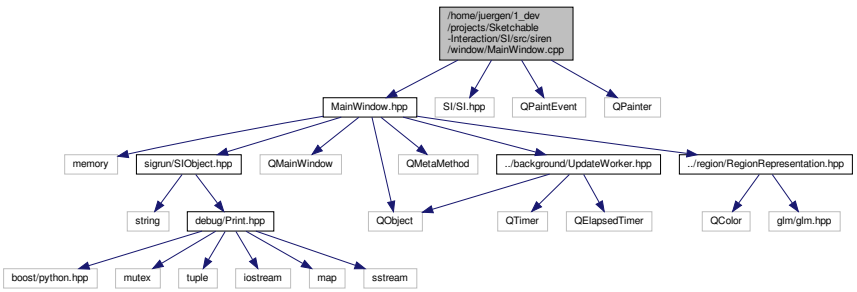

### Classes

- class GLSLProgram

## 5.83 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/window/MainWindow.cpp File Reference

```
#include "MainWindow.hpp"
#include <SI/SI.hpp>
#include <QPaintEvent>
#include <QPainter>
```

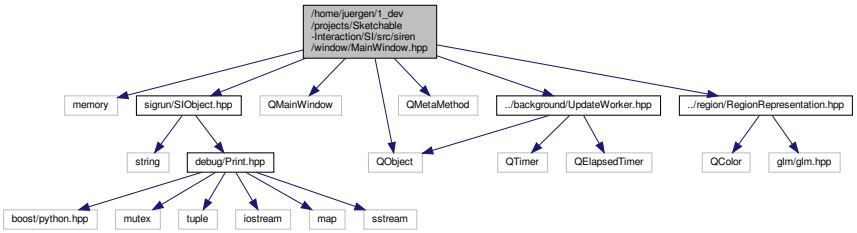
Include dependency graph for MainWindow.cpp:



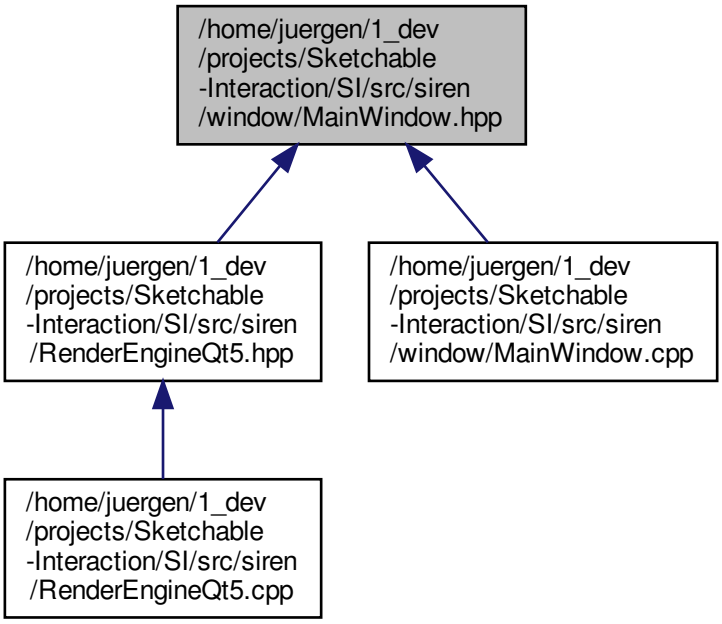
## 5.84 /home/juergen/1_dev/projects/Sketchable-Interaction/SI/src/siren/window/MainWindow.hpp File Reference

```
#include <memory>
#include <sigrun/SIObject.hpp>
#include <QMainWindow>
#include <QObject>
#include <QMetaMethod>
#include "../background/UpdateWorker.hpp"
#include "../region/RegionRepresentation.hpp"
```

Include dependency graph for MainWindow.hpp:




This graph shows which files directly or indirectly include this file:




### Classes

- class MainWindow

# Index