

Aqua

Generated by Doxygen 1.9.1

1 Class Index	1
1.1 Class List	1
2 Class Documentation	3
2.1 aq::Engine Class Reference	3
2.2 aq::Fish Class Reference	5
2.3 aq::Force Class Reference	7
2.4 aq::Island Class Reference	8
2.5 aq::Net::LocalisedIterator Class Reference	9
2.6 aq::Net Class Reference	10
2.7 GLSL::PerlinNoise Class Reference	12
2.7.1 Detailed Description	13
2.7.2 Member Function Documentation	14
2.7.2.1 colorFromHeight()	14
2.7.2.2 fractalNoise()	14
2.7.2.3 perlin()	14
2.7.2.4 randomGradient()	15
2.7.3 Member Data Documentation	15
2.7.3.1 u_top_left	15
2.8 aq::Net::Settings Struct Reference	15
Index	17

Chapter 1

Class Index

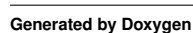
1.1 Class List

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

aq::Engine	3
aq::Fish	5
aq::Force	7
aq::Island	8
aq::Net::LocalisedIterator	9
aq::Net	10
GLSL::PerlinNoise	
Simple 2D perlin noise shader	12
aq::Net::Settings	15

Class Documentation

Collaboration diagram for aq::Engine:



Public Member Functions

- **Engine** (sf::Vector2u window_size)
- void **draw** ()
- void **startParalellLife** ()
- void **stopParalellLife** ()
- bool **isRunning** ()
- void **handeEvents** ()

Private Member Functions

- void **life** ()

Private Attributes

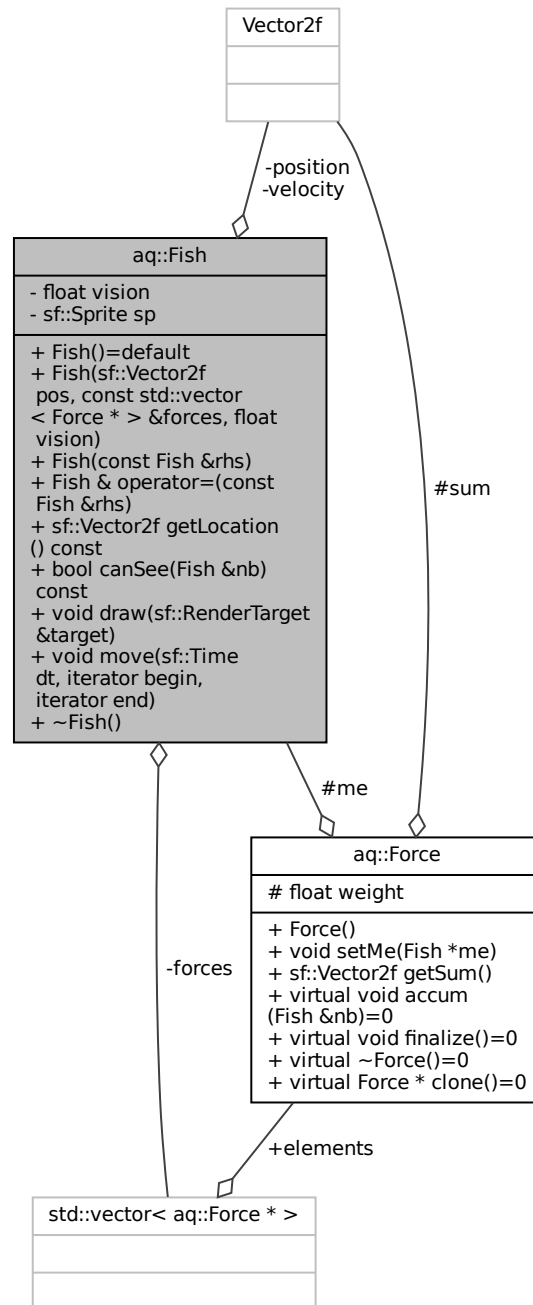
- sf::RenderWindow * **window**
- **Net** * **net**
- **Island** * **island**
- std::atomic< bool > **endLife**
- std::thread **bgLife**

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

- inc/engine.hpp
- src/engine.cpp

2.2 aq::Fish Class Reference

Collaboration diagram for aq::Fish:



Public Member Functions

- **Fish** (sf::Vector2f pos, const std::vector< [Force](#) * > &forces, float vision)
- **Fish** (const [Fish](#) &rhs)

- `Fish & operator= (const Fish &rhs)`
- `sf::Vector2f getLocation () const`
- `bool canSee (Fish &nb) const`
- `void draw (sf::RenderTarget &target)`
- `template<typename iterator >`
`void move (sf::Time dt, iterator begin, iterator end)`

Private Attributes

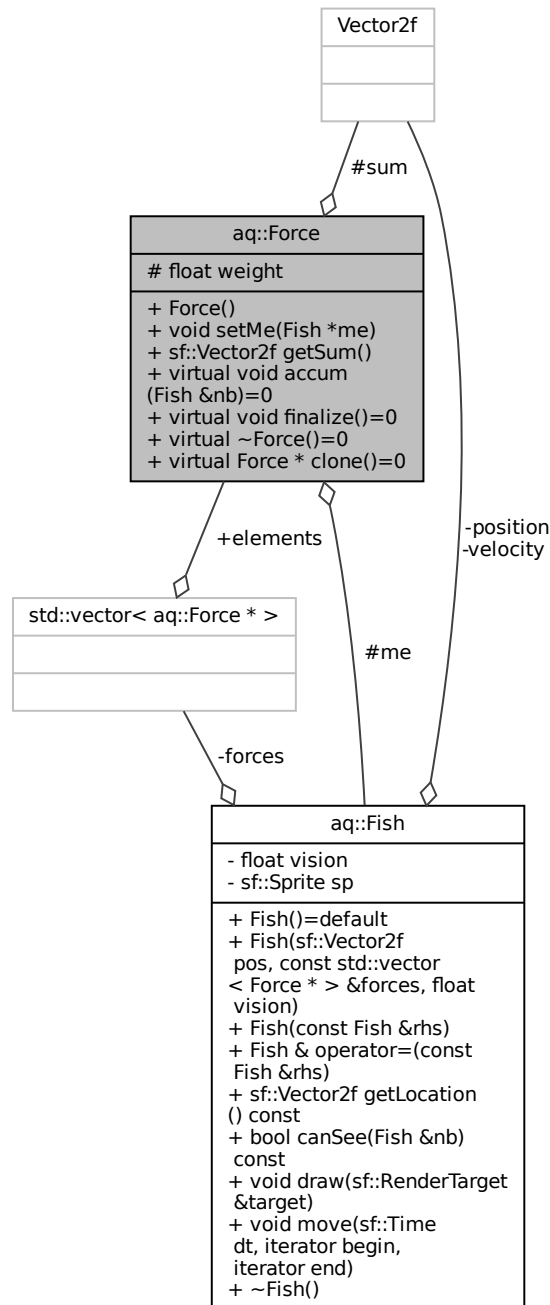
- `sf::Vector2f position`
- `sf::Vector2f velocity`
- `std::vector< Force * > forces`
- `float vision`
- `sf::Sprite sp`

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

- `inc/fish.hpp`
- `src/fish.cpp`

2.3 aq::Force Class Reference

Collaboration diagram for aq::Force:



Public Member Functions

- void **setMe** (Fish *me)
- sf::Vector2f **getSum** ()

- virtual void **accum** (Fish &nb)=0
- virtual void **finalize** ()=0
- virtual Force * **clone** ()=0

Protected Attributes

- Fish * **me**
- sf::Vector2f **sum**
- float **weight**

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

- inc/force.hpp
- src/force.cpp

2.4 aq::Island Class Reference

Collaboration diagram for aq::Island:



Public Member Functions

- **Island** (sf::Vector2u mapSize)
- void **draw** (sf::RenderTarget &target)

Private Attributes

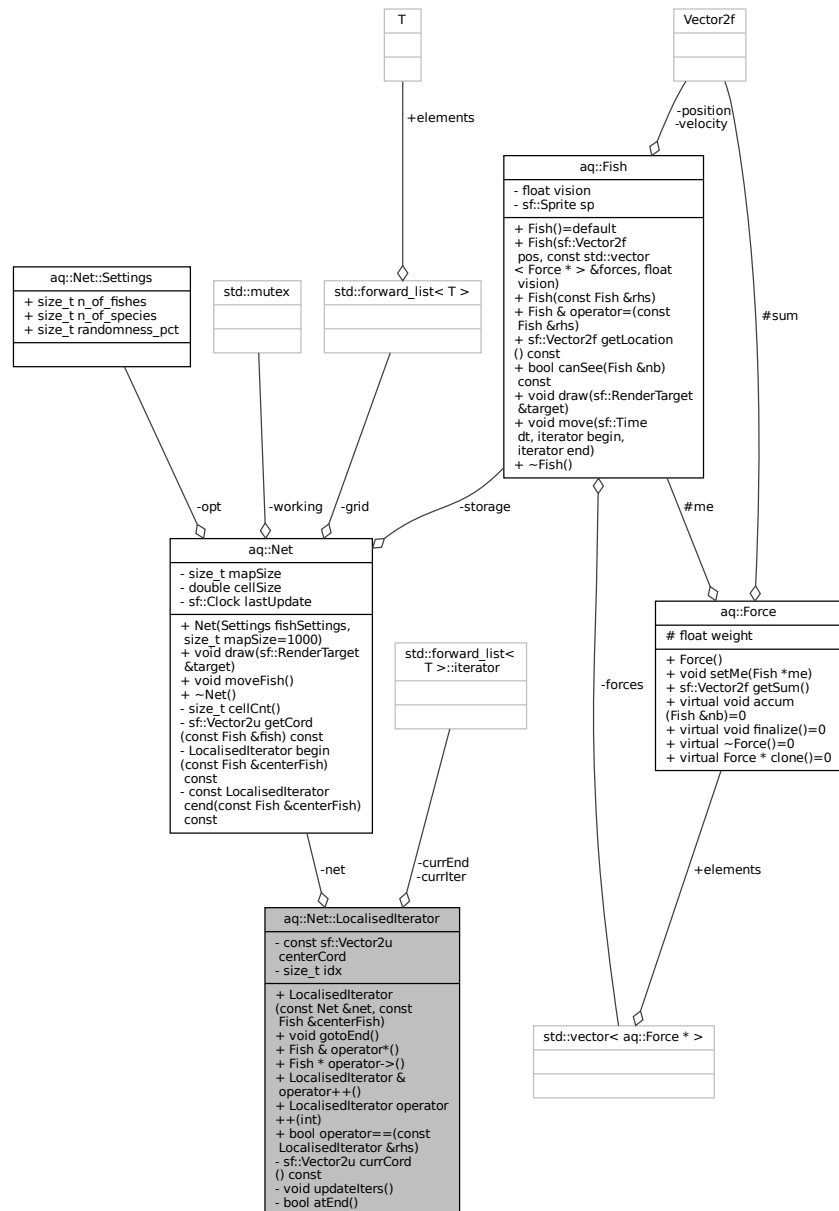
- sf::Sprite **canvasS**
- sf::Texture **canvasT**
- sf::Shader **shader**
- sf::Vector2u **mapSize**

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

- inc/island.hpp
- src/island.cpp

2.5 aq::Net::LocalisedIterator Class Reference

Collaboration diagram for aq::Net::LocalisedIterator:



Public Member Functions

- **LocalisedIterator** (const [Net](#) &net, const [Fish](#) ¢erFish)
- void **gotoEnd** ()
- [Fish](#) & **operator*** ()
- [Fish](#) * **operator->** ()
- [LocalisedIterator](#) & **operator++** ()
- [LocalisedIterator](#) **operator++** (int)
- bool **operator==** (const [LocalisedIterator](#) &rhs)

Classes

- class [LocalisedIterator](#)
- struct [Settings](#)

Public Types

- typedef std::forward_list< [Fish](#) * > **cell**

Public Member Functions

- **Net** ([Settings](#) fishSettings, size_t mapSize=1000)
- void **draw** (sf::RenderTarget &target)
- void **moveFish** ()

Private Member Functions

- size_t **cellCnt** ()
- sf::Vector2u **getCord** (const [Fish](#) &fish) const
- [LocalisedIterator](#) **begin** (const [Fish](#) ¢erFish) const
- const [LocalisedIterator](#) **cend** (const [Fish](#) ¢erFish) const

Private Attributes

- const [Settings](#) **opt**
- [Fish](#) * **storage**
- cell ** **grid**
- size_t **mapSize**
- double **cellSize**
- sf::Clock **lastUpdate**
- std::mutex **working**

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

- inc/net.hpp
- src/net.cpp

2.7 GLSL::PerlinNoise Class Reference

Simple 2D perlin noise shader.

Collaboration diagram for GLSL::PerlinNoise:

GLSL::PerlinNoise
<ul style="list-style-type: none"> + uniform vec2 u_seed + uniform int u_octaves + uniform float u_gridSize + uniform float u_amplitude + uniform float u_water_level + uniform float u_sand_level + uniform float u_bw_mode + uniform vec4 col_low_water + uniform vec4 col_high_water + uniform vec4 col_low_sand and 6 more...
<ul style="list-style-type: none"> + float interpolate(float a, float b, float w) + float cap(float value) + vec2 randomGradient(ivec2 cord) + float dotGridGradient(ivec2 cord, vec2 pos) + float perlin(vec2 pos) + float fractalNoise(vec2 pos) + vec4 colorFromHeight(float height) + void main()

Public Member Functions

- float [interpolate](#) (float a, float b, float w)
Smoothly interpolates between two values.
- float [cap](#) (float value)
Caps a value between [0, 1].
- vec2 [randomGradient](#) (ivec2 cord)
Computes a pseudo random gradient vector for a given integer coordinate.
- float [dotGridGradient](#) (ivec2 cord, vec2 pos)
Computes the dot product of a random gradient vector and a given position.
- float [perlin](#) (vec2 pos)

2D Perlin noise

- float `fractalNoise` (vec2 pos)
Computes a fractal sum of perlin noise.
- vec4 `colorFromHeight` (float height)
Computes a color based on the height.
- void `main` ()
Main function.

Public Attributes

- uniform vec2 `u_seed`
Seed used as offset.
- uniform int `u_octaves`
Number of patterns to sum.
- uniform float `u_gridSize`
Size of the grid.
- uniform float `u_amplitude`
Start amplitude of the noise.
- uniform float `u_water_level`
Threshold for water [0, 1].
- uniform float `u_sand_level`
Threshold for sand [0, 1].
- uniform float `u_bw_mode`
B&W mask mode toggle, 0 or 1.
- uniform vec4 `col_low_water`
Color for deep water.
- uniform vec4 `col_high_water`
Color for shallow water.
- uniform vec4 `col_low_sand`
Color for low sand.
- uniform vec4 `col_high_sand`
Color for high sand.
- uniform vec4 `col_low_grass`
Color for low grass.
- uniform vec4 `col_high_grass`
Color for high grass.
- uniform vec2 `u_resolution`
- s uniform vec2 `u_top_left`
< Size of the window
- uniform vec2 `u_bottom_right`
Bottom right corner of the visible area.

2.7.1 Detailed Description

Simple 2D perlin noise shader.

Code based on the the Perlin noise wikipedia page: https://en.wikipedia.org/wiki/Perlin_noise

Remarks

Fragment-Shader

2.7.2 Member Function Documentation

2.7.2.1 colorFromHeight()

```
vec4 GLSL::PerlinNoise::colorFromHeight (
    float height ) [inline]
```

Computes a color based on the height.

Parameters

<i>height</i>	in [0, 1]
---------------	-----------

2.7.2.2 fractalNoise()

```
float GLSL::PerlinNoise::fractalNoise (
    vec2 pos ) [inline]
```

Computes a fractal sum of perlin noise.

Returns

[0, 1]

2.7.2.3 perlin()

```
float GLSL::PerlinNoise::perlin (
    vec2 pos ) [inline]
```

2D Perlin noise

Parameters

<i>pos</i>	Position in 2D space
------------	----------------------

Returns

[-1, 1]

2.7.2.4 randomGradient()

```
vec2 GLSL::PerlinNoise::randomGradient (
    ivec2 cord ) [inline]
```

Computes a pseudo random gradient vector for a given integer coordinate.

Returns

Vector with length 1

2.7.3 Member Data Documentation

2.7.3.1 u_top_left

```
s uniform vec2 GLSL::PerlinNoise::u_top_left
```

< Size of the window

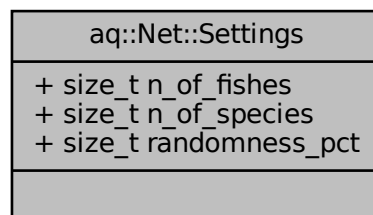
Top left corner of the visible area

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

- src/perlin.frag

2.8 aq::Net::Settings Struct Reference

Collaboration diagram for aq::Net::Settings:



Public Attributes

- size_t **n_of_fishes** = 100
- size_t **n_of_species** = 1
- size_t **randomness_pct** = 0

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

- inc/net.hpp

Index

- aq::Engine, [3](#)
- aq::Fish, [5](#)
- aq::Force, [7](#)
- aq::Island, [8](#)
- aq::Net, [10](#)
- aq::Net::LocalisedIterator, [9](#)
- aq::Net::Settings, [15](#)

- colorFromHeight
 - GLSL::PerlinNoise, [14](#)

- fractalNoise
 - GLSL::PerlinNoise, [14](#)

- GLSL::PerlinNoise, [12](#)
 - colorFromHeight, [14](#)
 - fractalNoise, [14](#)
 - perlin, [14](#)
 - randomGradient, [14](#)
 - u_top_left, [15](#)

- perlin
 - GLSL::PerlinNoise, [14](#)

- randomGradient
 - GLSL::PerlinNoise, [14](#)

- u_top_left
 - GLSL::PerlinNoise, [15](#)