Aqua

Generated by Doxygen 1.9.1

Chapter 1

Class Index

1.1 Class List

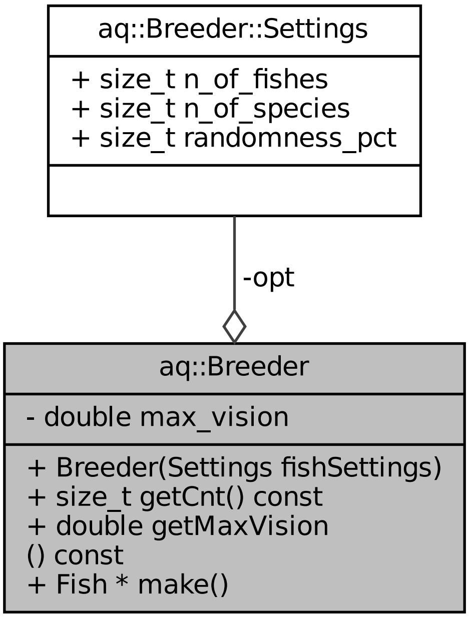
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 aq::Breeder Class Reference

Collaboration diagram for aq::Breeder:



Classes

- struct Settings

Public Member Functions

- **Breeder** (Settings fishSettings)
- size_t **getCnt** () const
- double **getMaxVision** () const
- Fish ∗ **make** ()

Private Attributes

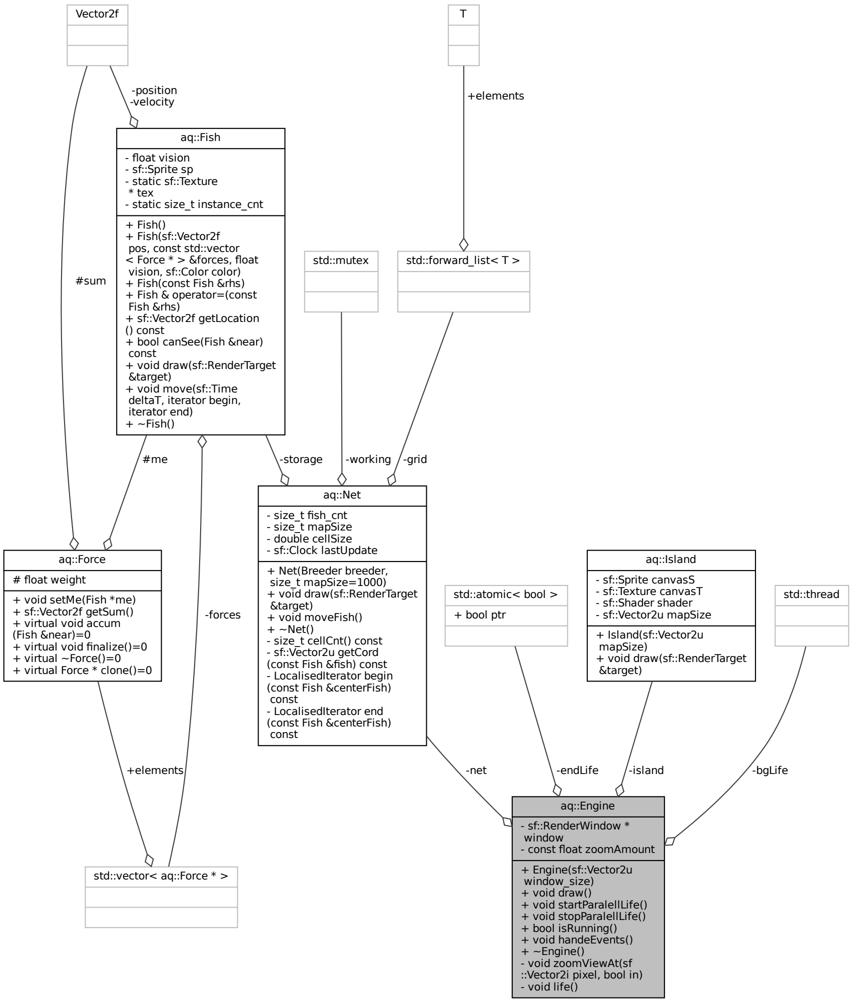
- const Settings **opt**
- double **max_vision** = 0

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

- inc/breeder.hpp
- src/breeder.cpp

2.2 aq::Engine Class Reference

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 **zoomViewAt** (sf::Vector2i pixel, bool in)
- void **life** ()

Private Attributes

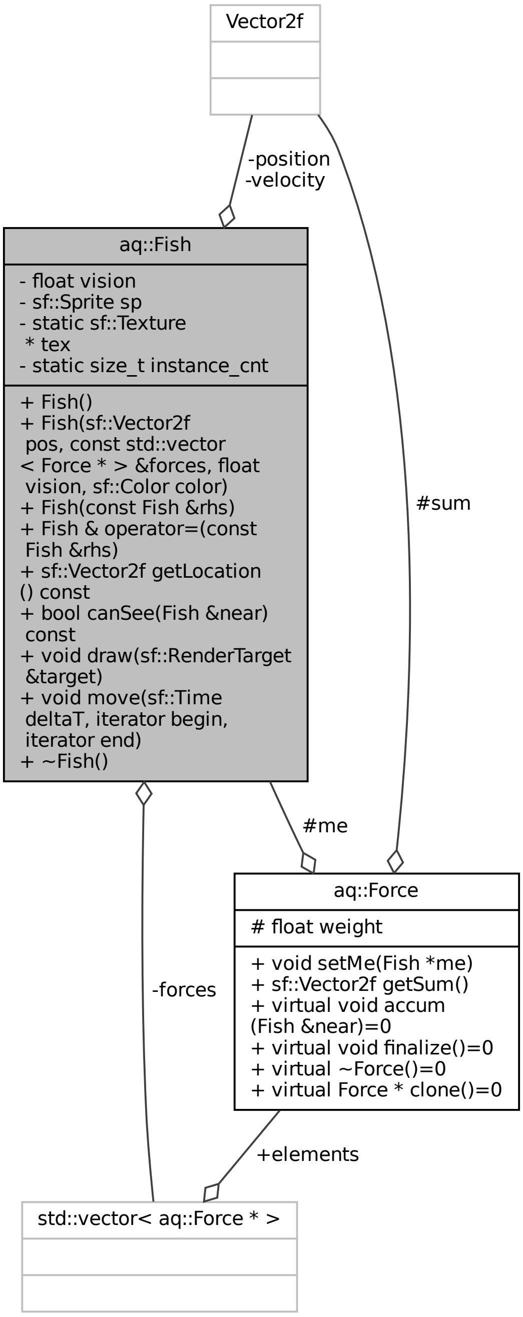
- sf::RenderWindow ∗ **window**
- Net ∗ **net**
- Island ∗ **island**
- std::atomic< bool > **endLife**
- const float **zoomAmount** = 1.3f
- std::thread **bgLife**

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

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

2.3 aq::Fish Class Reference

Collaboration diagram for aq::Fish:



Public Member Functions

- **Fish** (sf::Vector2f pos, const std::vector$<$ Force $*$ $>$ &forces, float vision, sf::Color color)
- **Fish** (const Fish &rhs)

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

Private Attributes

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

Static Private Attributes

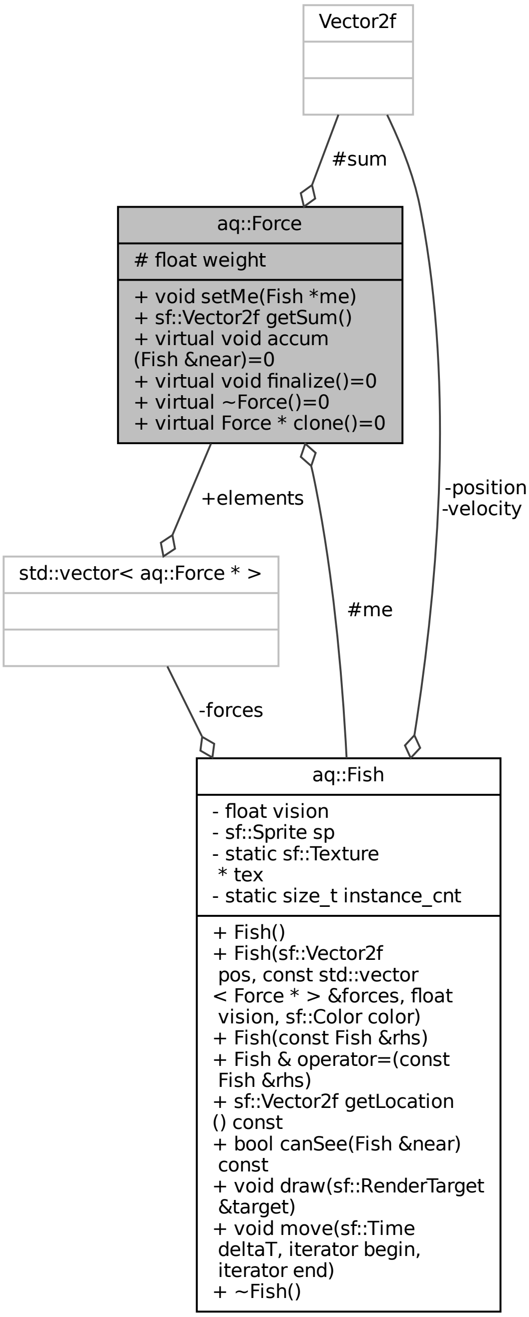
- static sf::Texture * **tex** = nullptr
- static size_t **instance_cnt** = 0

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

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

2.4 aq::Force Class Reference

Collaboration diagram for aq::Force:



Public Member Functions

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

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

Protected Attributes

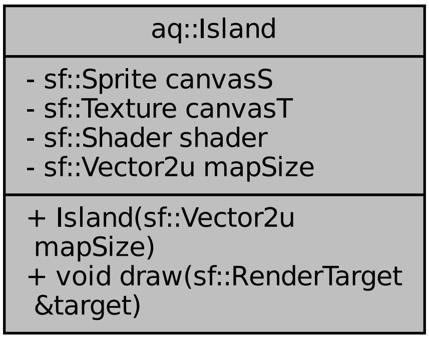
- Fish ∗ **me** {nullptr}
- sf::Vector2f **sum**
- float **weight** {0}

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

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

2.5 aq::Island Class Reference

Collaboration diagram for aq::Island:



Public Member Functions

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

Private Attributes

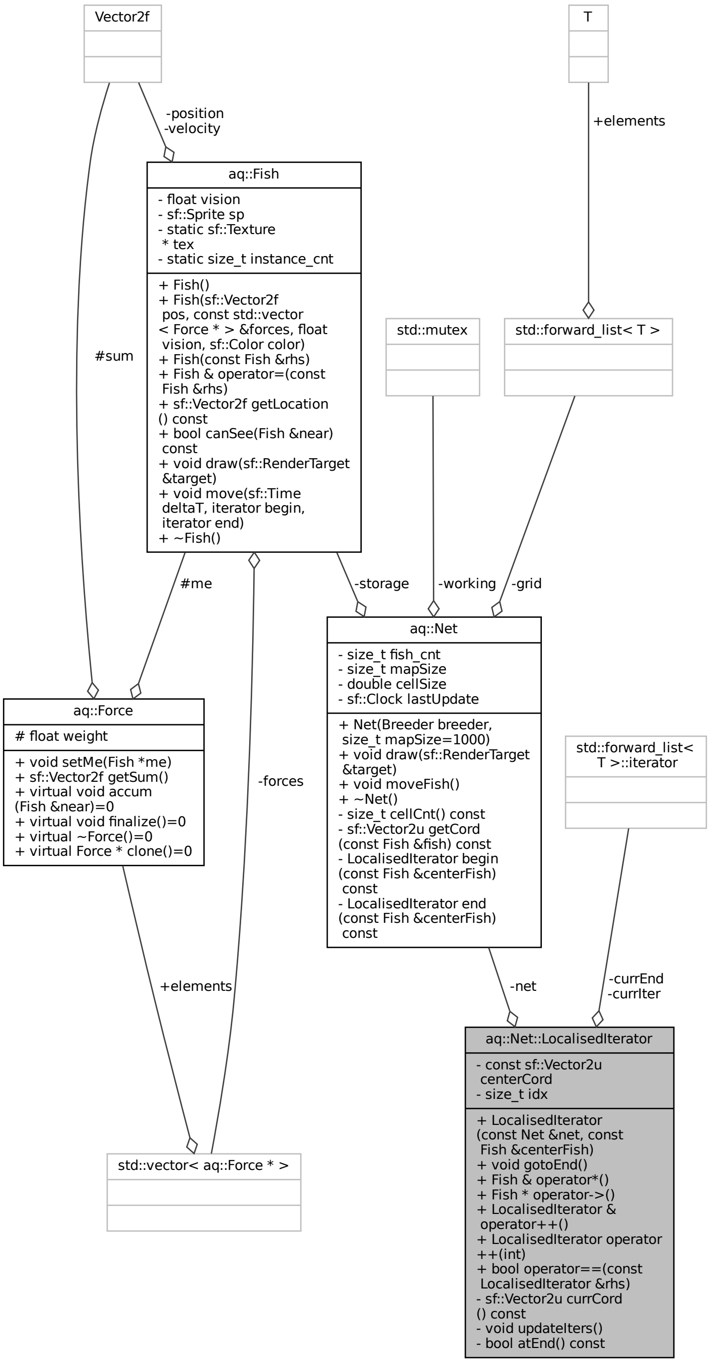
- sf::Sprite **canvasS**
- sf::Texture **canvasT**
- sf::Shader **shader**
- sf::Vector2u **mapSize**

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

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

2.6 aq::Net::LocalisedIterator Class Reference

Collaboration diagram for aq::Net::LocalisedIterator:



Public Member Functions

- **LocalisedIterator** (const Net &net, const Fish &centerFish)
- void **gotoEnd** ()

- Fish & **operator∗** ()
- Fish ∗ **operator->** ()
- LocalisedIterator & **operator++** ()
- LocalisedIterator **operator++** (int)
- bool **operator==** (const LocalisedIterator &rhs)

Private Member Functions

- sf::Vector2u **currCord** () const
- void **updateIters** ()
- bool **atEnd** () const

Private Attributes

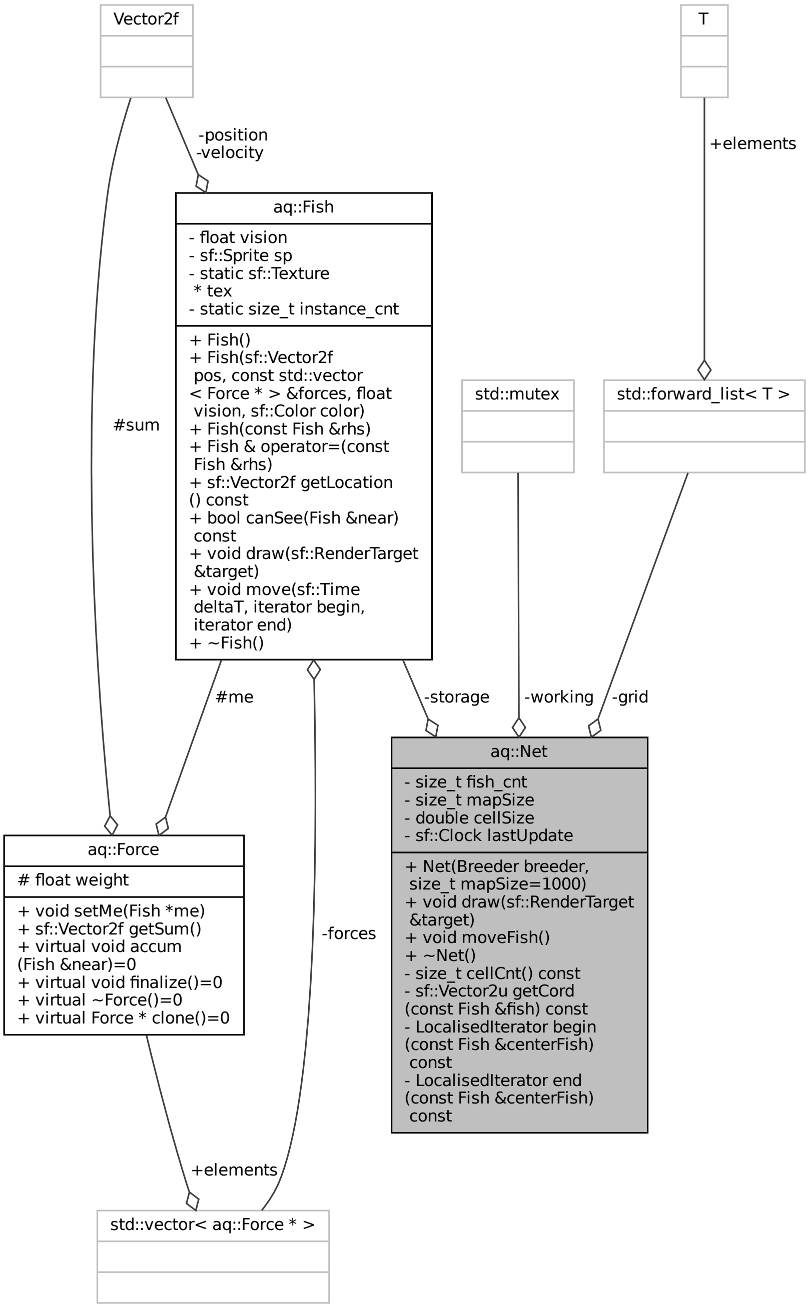
- const Net & **net**
- const sf::Vector2u **centerCord**
- cell::iterator **currIter**
- cell::iterator **currEnd**
- size_t **idx** {0}

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

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

2.7 aq::Net Class Reference

Collaboration diagram for aq::Net:



Classes

- class LocalisedIterator

Public Types

- typedef std::forward_list< Fish * > **cell**

Public Member Functions

- **Net** (Breeder breeder, size_t mapSize=1000)
- void **draw** (sf::RenderTarget &target)
- void **moveFish** ()

Private Member Functions

- size_t **cellCnt** () const
- sf::Vector2u **getCord** (const Fish &fish) const
- LocalisedIterator **begin** (const Fish &centerFish) const
- LocalisedIterator **end** (const Fish &centerFish) const

Private Attributes

- size_t **fish_cnt**
- Fish * **storage**
- cell ** **grid**
- size_t **mapSize**
- double **cellSize**
- sf::Clock **lastUpdate**
- std::mutex **working**

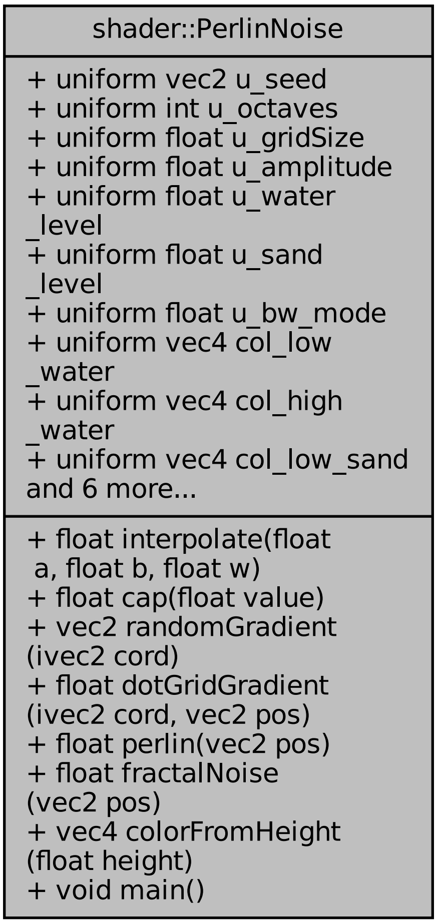
The documentation for this class was generated from the following files:

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

2.8 shader::PerlinNoise Class Reference

Simple 2D perlin noise shader.

Collaboration diagram for shader::PerlinNoise:


shader::PerlinNoise
+ 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...
+ 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 amlitude 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

  Size of the window.
- uniform vec2 u_top_left

  Top left corner of the visible area.
- uniform vec2 u_bottom_right

  Bottom right corner of the visible area.

2.8.1 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

2.8.2 Member Function Documentation

2.8.2.1 colorFromHeight()

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

Computes a color based on the height.

Parameters

height	in [0, 1]

2.8.2.2 fractalNoise()

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

Computes a fractal sum of perlin noise.

Returns

[0, 1]

2.8.2.3 perlin()

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

2D Perlin noise

Parameters

pos	Position in 2D space

Returns

[-1, 1]

2.8.2.4 randomGradient()

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

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

Returns

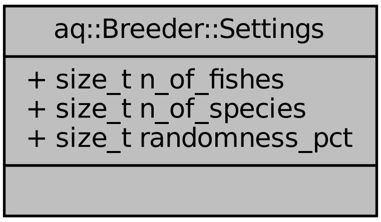
Vector with length 1

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

- src/perlin.frag

2.9 aq::Breeder::Settings Struct Reference

Collaboration diagram for aq::Breeder::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/breeder.hpp

Index