

Aqua

Generated by Doxygen 1.9.1

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 Class Documentation	5
3.1 aq::AlignmentForce Class Reference	5
3.2 aq::Breeder Class Reference	7
3.3 aq::CohesionForce Class Reference	8
3.4 aq::Breeder::Dependency Struct Reference	10
3.5 aq::Engine Class Reference	11
3.6 aq::Fish Class Reference	13
3.7 aq::Force Class Reference	14
3.8 aq::Island Class Reference	16
3.9 aq::IslandForce Class Reference	18
3.9.1 Member Data Documentation	20
3.9.1.1	20
3.10 aq::Net::LocalisedIterator Class Reference	21
3.11 aq::Island::Map Struct Reference	22
3.11.1 Member Function Documentation	23
3.11.1.1 operator()()	23
3.12 aq::MinSpeedForce Class Reference	24
3.13 aq::MouseForce Class Reference	26
3.14 aq::Net Class Reference	29
3.15 shader::PerlinNoise Class Reference	30
3.15.1 Detailed Description	33
3.15.2 Member Function Documentation	33
3.15.2.1 colorFromHeight()	33
3.15.2.2 fractalNoise()	33
3.15.2.3 perlin()	33
3.15.2.4 randomGradient()	34
3.16 aq::SeparationForce Class Reference	35
3.17 aq::Breeder::Settings Struct Reference	37
3.18 aq::WaterResistantForce Class Reference	38
Index	41

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

aq::Breeder	7
aq::Breeder::Dependency	10
aq::Engine	11
aq::Fish	13
aq::Force	14
aq::AlignmentForce	5
aq::CohesionForce	8
aq::IslandForce	18
aq::MinSpeedForce	24
aq::MouseForce	26
aq::SeparationForce	35
aq::WaterResistantForce	38
aq::Island	16
aq::Net::LocalisedIterator	21
aq::Island::Map	22
aq::Net	29
shader::PerlinNoise	30
aq::Breeder::Settings	37

Chapter 2

Class Index

2.1 Class List

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

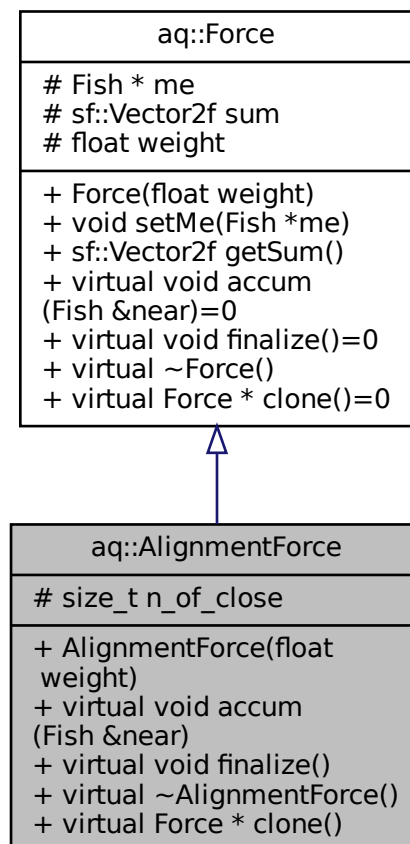
aq::AlignmentForce	5
aq::Breeder	7
aq::CohesionForce	8
aq::Breeder::Dependency	10
aq::Engine	11
aq::Fish	13
aq::Force	14
aq::Island	16
aq::IslandForce	18
aq::Net::LocalisedIterator	21
aq::Island::Map	22
aq::MinSpeedForce	24
aq::MouseForce	26
aq::Net	29
shader::PerlinNoise	
Simple 2D perlin noise shader	30
aq::SeparationForce	35
aq::Breeder::Settings	37
aq::WaterResistantForce	38

Chapter 3

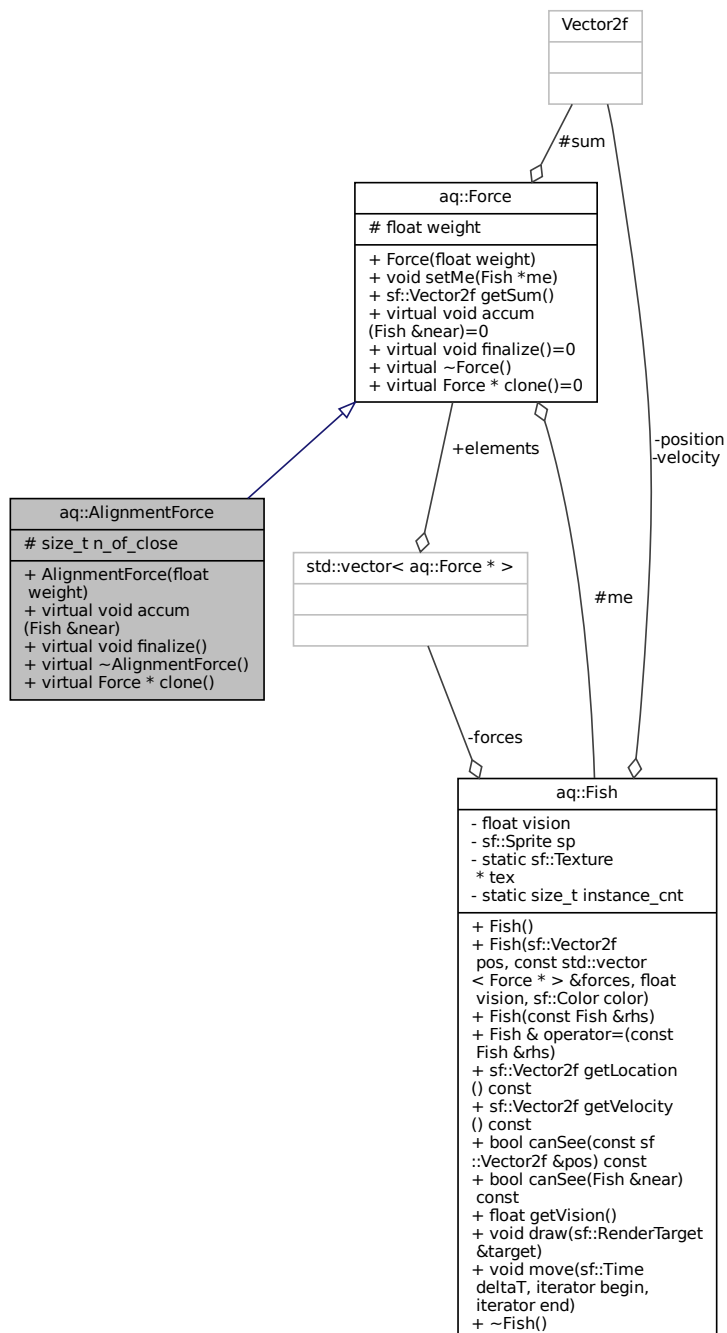
Class Documentation

3.1 aq::AlignmentForce Class Reference

Inheritance diagram for aq::AlignmentForce:



Collaboration diagram for `aq::AlignmentForce`:



Public Member Functions

- **AlignmentForce** (float weight)
- virtual void **accum** (Fish &near)
- virtual void **finalize** ()
- virtual Force * **clone** ()

Protected Attributes

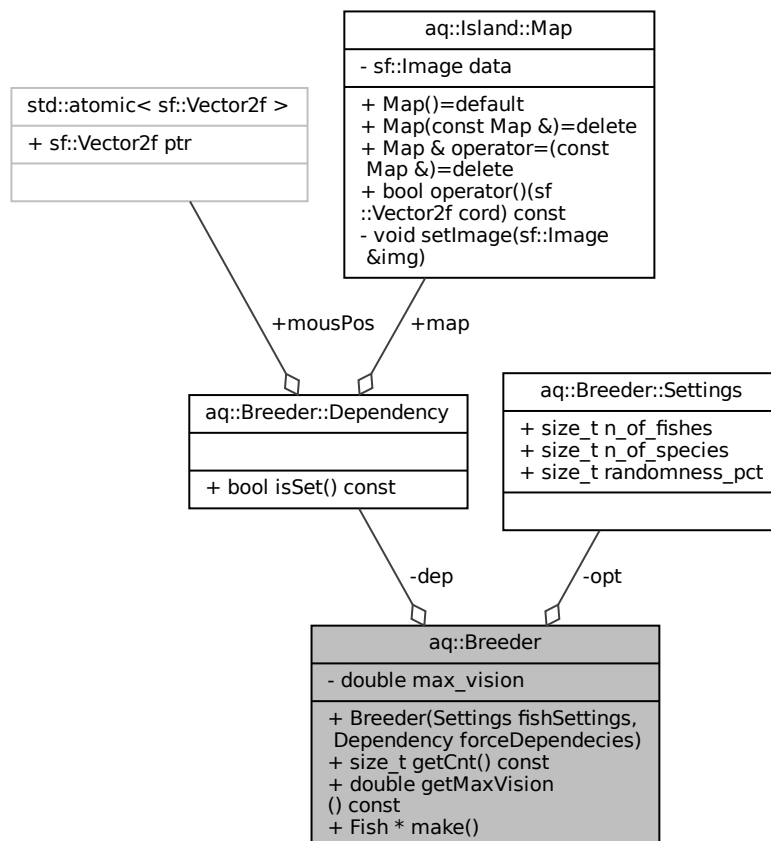
- `size_t n_of_close {0}`

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

- `inc/forces.hpp`

3.2 aq::Breeder Class Reference

Collaboration diagram for aq::Breeder:



Classes

- struct [Dependency](#)
- struct [Settings](#)

Public Member Functions

- **Breeder** ([Settings](#) fishSettings, [Dependency](#) forceDependencies)
- `size_t` **getCnt** () const
- `double` **getMaxVision** () const
- [Fish](#) * **make** ()

Private Attributes

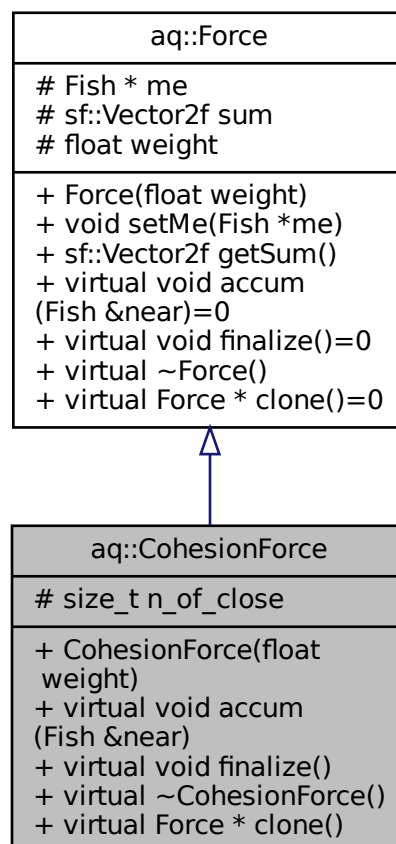
- `const` [Settings](#) **opt**
- `const` [Dependency](#) **dep**
- `double` **max_vision** = 0

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

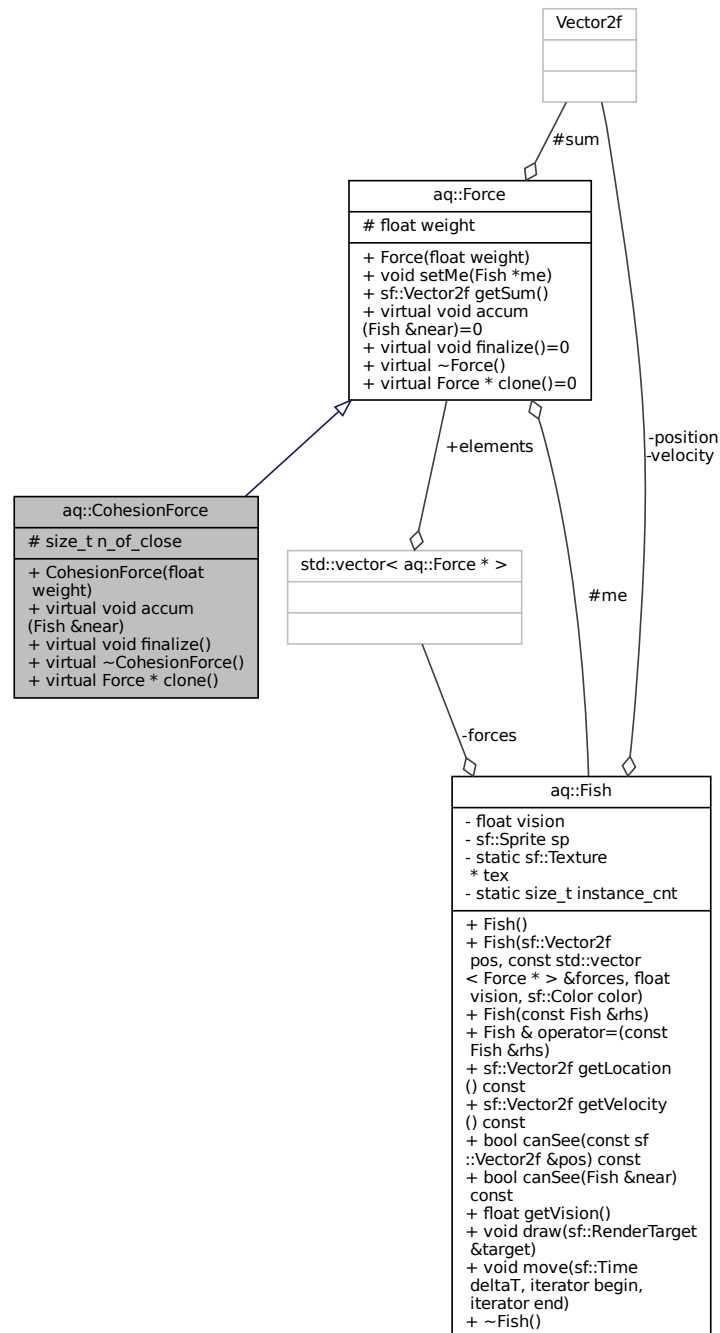
- inc/breeder.hpp
- src/breeder.cpp

3.3 aq::CohesionForce Class Reference

Inheritance diagram for aq::CohesionForce:



Collaboration diagram for aq::CohesionForce:



Public Member Functions

- **CohesionForce** (float weight)
- virtual void **accum** ([Fish](#) &near)
- virtual void **finalize** ()
- virtual [Force](#) * **clone** ()

Protected Attributes

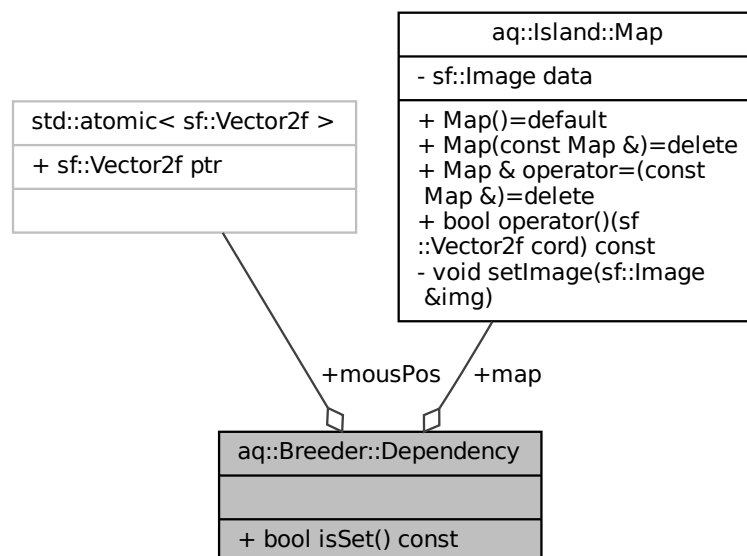
- `size_t n_of_close {0}`

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

- `inc/forces.hpp`

3.4 aq::Breeder::Dependency Struct Reference

Collaboration diagram for aq::Breeder::Dependency:



Public Member Functions

- `bool isSet () const`

Public Attributes

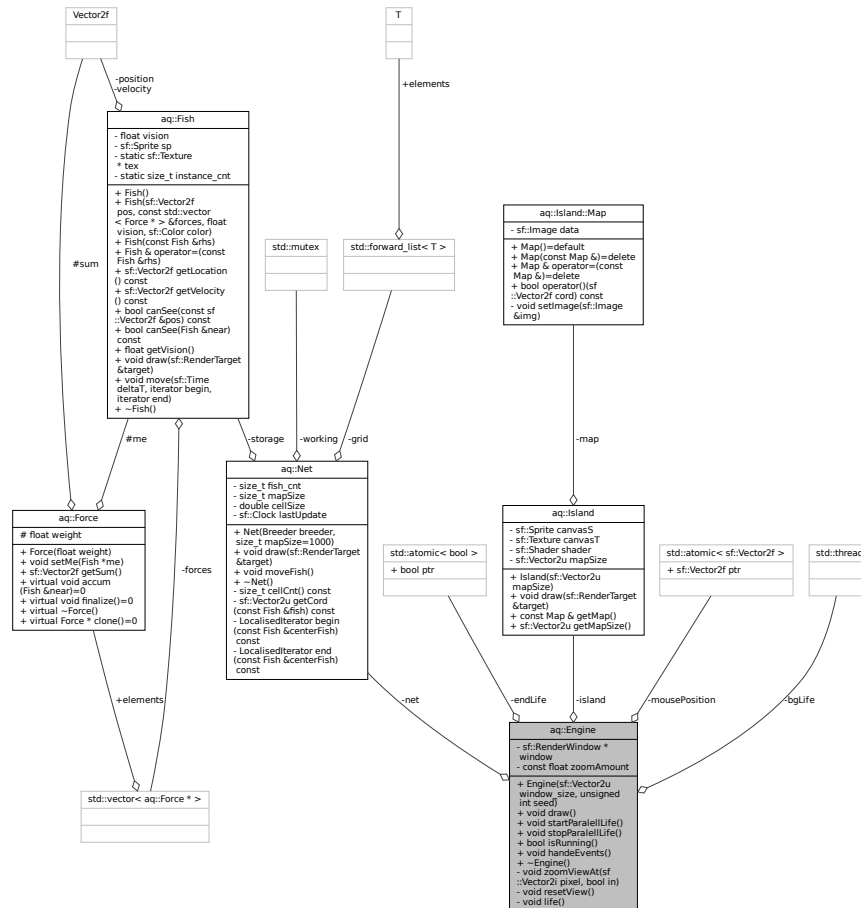
- `const Island::Map * map`
- `const std::atomic< sf::Vector2f > * mousPos`

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

- `inc/breeder.hpp`

3.5 aq::Engine Class Reference

Collaboration diagram for aq::Engine:



Public Member Functions

- **Engine** (sf::Vector2u window_size, unsigned int seed)
- void **draw** ()
- void **startParallelLife** ()
- void **stopParallelLife** ()
- bool **isRunning** ()
- void **handleEvents** ()

Private Member Functions

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

Private Attributes

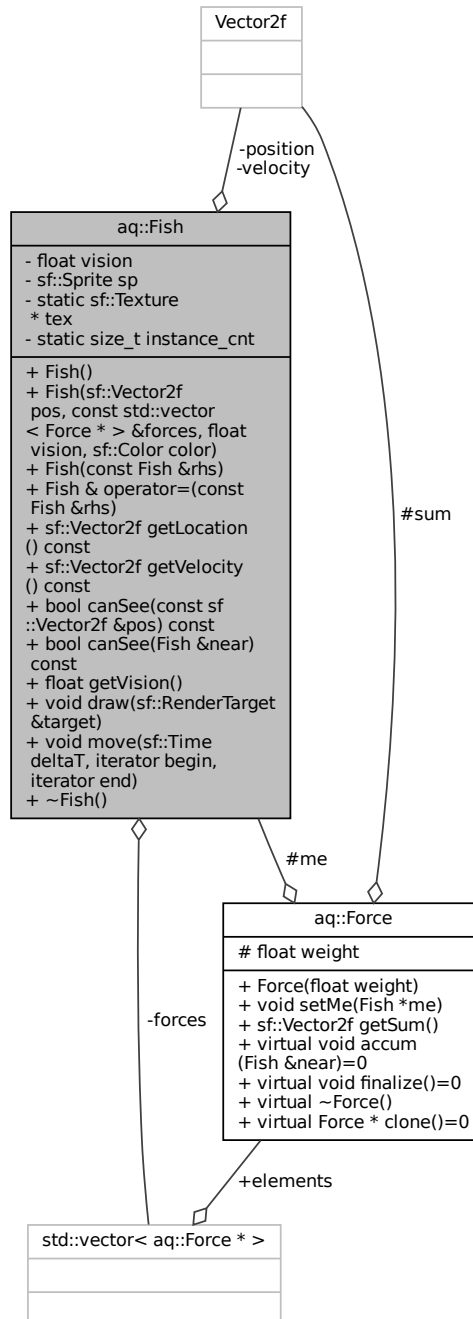
- `sf::RenderWindow * window`
- `Net * net`
- `Island * island`
- `std::atomic< bool > endLife`
- `const float zoomAmount = 1.3F`
- `std::thread bgLife`
- `std::atomic< sf::Vector2f > mousePosition`

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

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

3.6 aq::Fish Class Reference

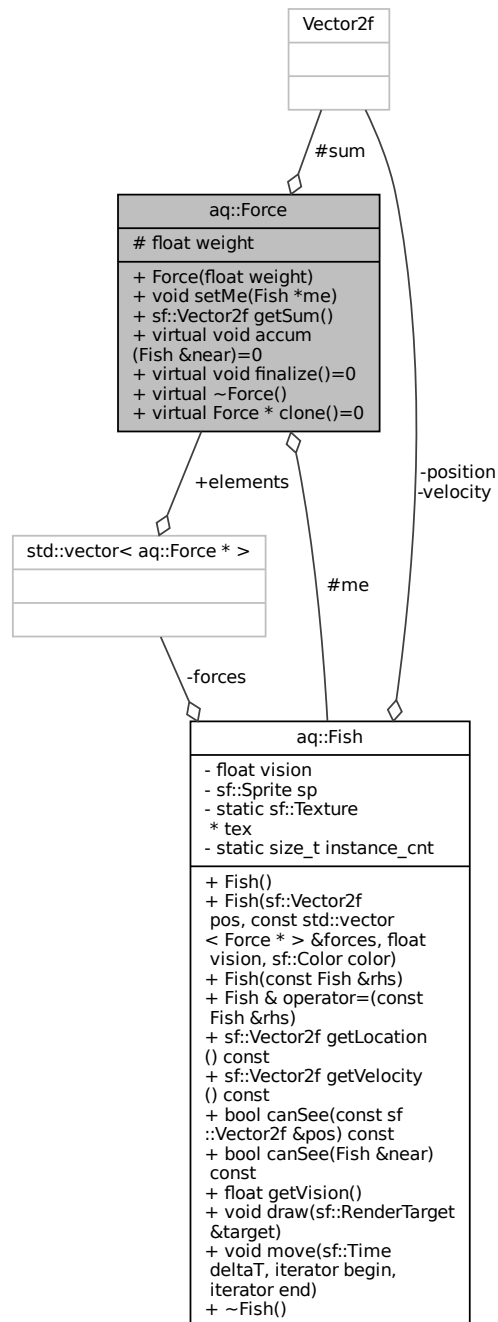
Collaboration diagram for aq::Fish:



Public Member Functions

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

Collaboration diagram for aq::Force:



Public Member Functions

- **Force** (float weight)
- void **setMe** (Fish *me)
- sf::Vector2f **getSum** ()
- virtual void **accum** (Fish &near)=0
- virtual void **finalize** ()=0
- virtual **Force** * **clone** ()=0

Protected Attributes

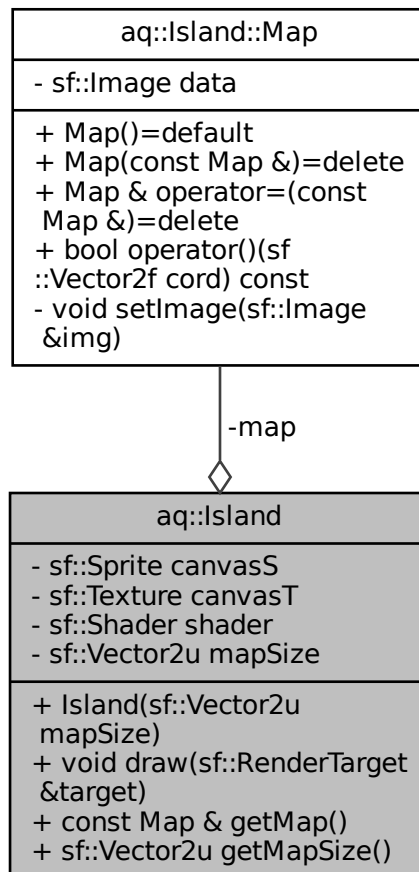
- `Fish * me {nullptr}`
- `sf::Vector2f sum {0, 0}`
- `float weight`

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

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

3.8 aq::Island Class Reference

Collaboration diagram for aq::Island:



Classes

- struct `Map`

Public Member Functions

- **Island** (sf::Vector2u mapSize)
- void **draw** (sf::RenderTarget &target)
- const **Map** & **getMap** ()
- sf::Vector2u **getMapSize** ()

Private Attributes

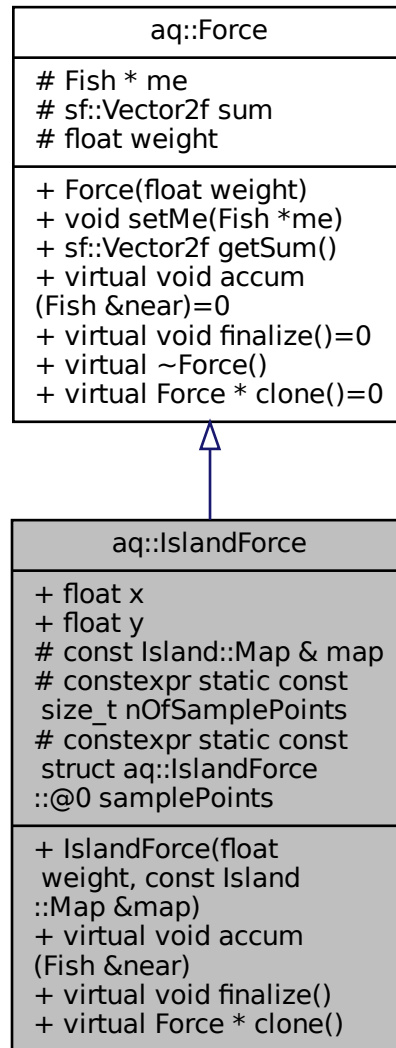
- sf::Sprite **canvasS**
- sf::Texture **canvasT**
- sf::Shader **shader**
- sf::Vector2u **mapSize**
- **Map** **map**

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

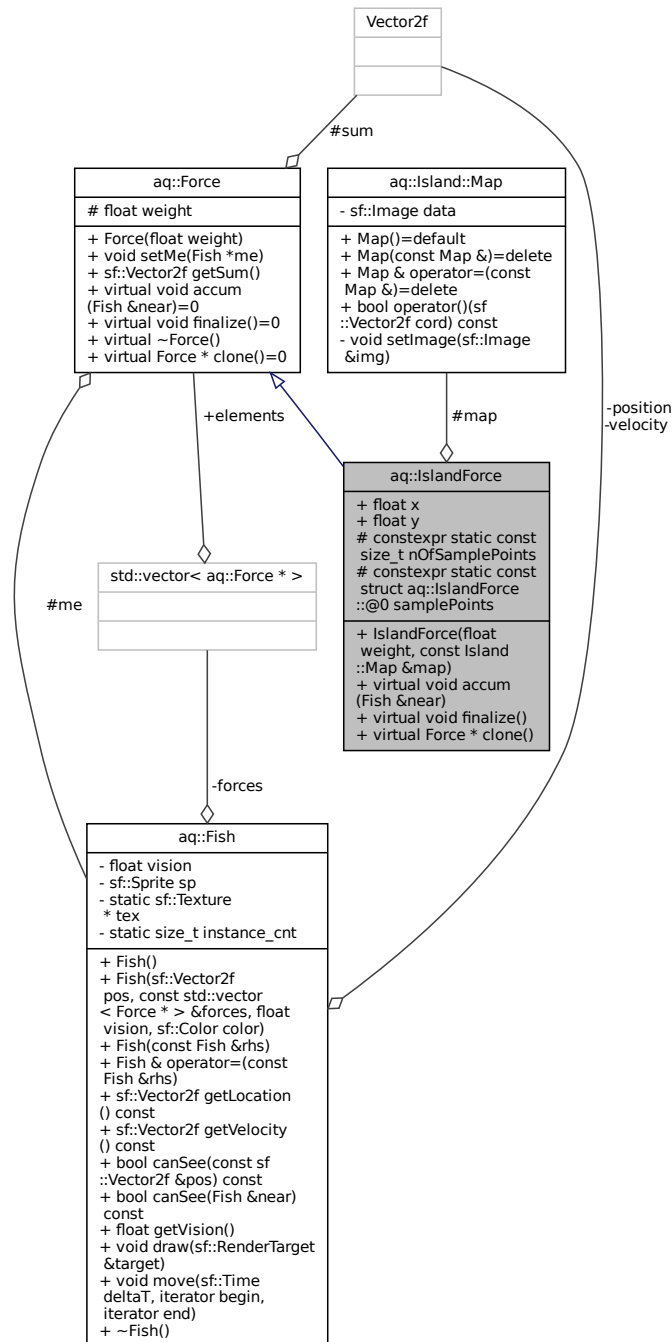
- inc/island.hpp
- src/island.cpp

3.9 aq::IslandForce Class Reference

Inheritance diagram for aq::IslandForce:



Collaboration diagram for aq::IslandForce:



Public Member Functions

- **IslandForce** (float weight, const [Island::Map](#) &map)
- virtual void **accum** ([Fish](#) &near)
- virtual void **finalize** ()
- virtual [Force](#) * **clone** ()

Protected Attributes

- const [Island::Map](#) & map

Static Protected Attributes

- constexpr static const size_t nOfSamplePoints = 36
- struct {
 - float x
 - float y
 } samplePoints [nOfSamplePoints]

3.9.1 Member Data Documentation

3.9.1.1

```
constexpr { ... } aq::IslandForce::samplePoints[nOfSamplePoints] [static], [protected]
```

Initial value:

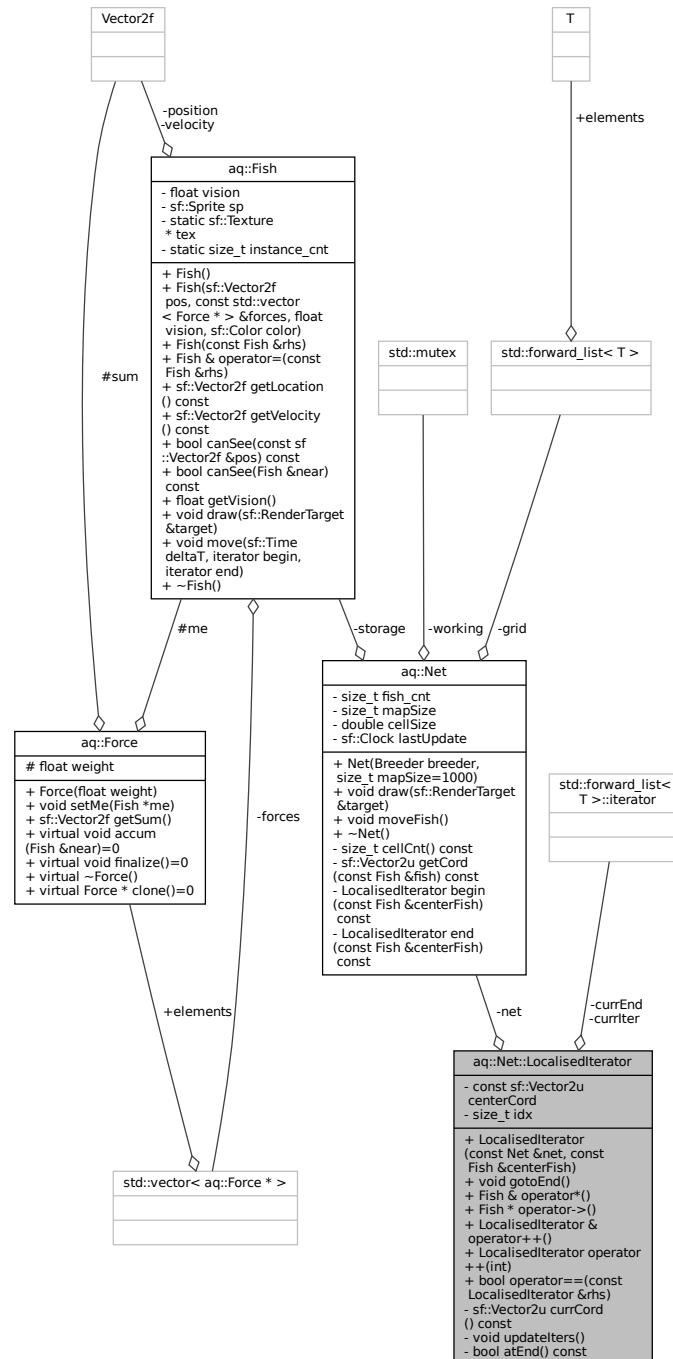
=

```
{ {1.000, 0.000}, {0.940, 0.342}, {0.766, 0.643}, {0.500, 0.866}, {0.174, 0.985}, {-0.174, 0.985},
  {-0.500, 0.866}, {-0.766, 0.643}, {-0.940, 0.342}, {-1.000, 0.000}, {-0.940, -0.342}, {-0.766,
  -0.643}, {-0.500, -0.866}, {-0.174, -0.985}, {0.174, -0.985}, {0.500, -0.866}, {0.766, -0.643},
  {0.940, -0.342}, {0.667, 0.000}, {0.577, 0.333}, {0.333, 0.577}, {0.000, 0.667}, {-0.333, 0.577},
  {-0.577, 0.333}, {-0.667, 0.000}, {-0.577, -0.333}, {-0.333, -0.577}, {-0.000, -0.667}, {0.333,
  -0.577}, {0.577, -0.333}, {0.333, 0.000}, {0.167, 0.289}, {-0.167, 0.289}, {-0.333, 0.000}, {-0.167,
  -0.289}, {0.167, -0.289}}
```

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

- inc/forces.hpp

Collaboration diagram for `aq::Net::LocalisedIterator`:



- **LocalisedIterator** (const **Net** &net, const **Fish** ¢erFish)
- 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

3.11 aq::Island::Map Struct Reference

Collaboration diagram for aq::Island::Map:

aq::Island::Map
- sf::Image data
+ Map()=default + Map(const Map &)=delete + Map & operator=(const Map &)=delete + bool operator()(sf::Vector2f cord) const - void setImage(sf::Image &img)

Public Member Functions

- **Map** (const [Map](#) &)=delete
- [Map](#) & **operator=** (const [Map](#) &)=delete
- bool **operator()** (sf::Vector2f cord) const
Can fish go to cord.

Private Member Functions

- void **setImage** (sf::Image &img)

Private Attributes

- sf::Image **data**

Friends

- class **Island**

3.11.1 Member Function Documentation

3.11.1.1 operator()

```
bool Island::Map::operator() (
    sf::Vector2f cord ) const
```

Can fish go to cord.

Parameters

<i>cord</i>	cord on map
-------------	-------------

Returns

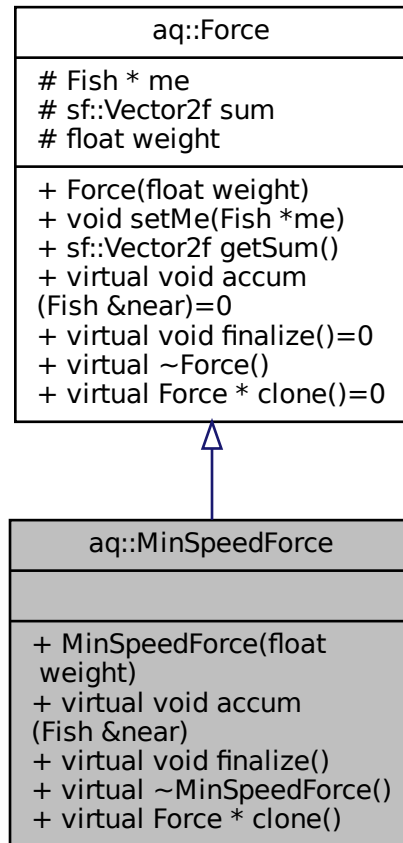
true if water, false is island

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

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

3.12 aq::MinSpeedForce Class Reference

Inheritance diagram for aq::MinSpeedForce:



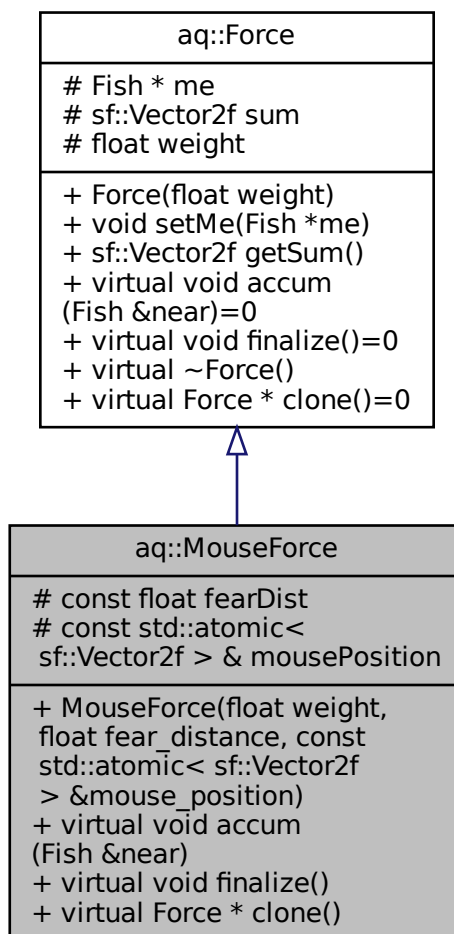
Additional Inherited Members

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

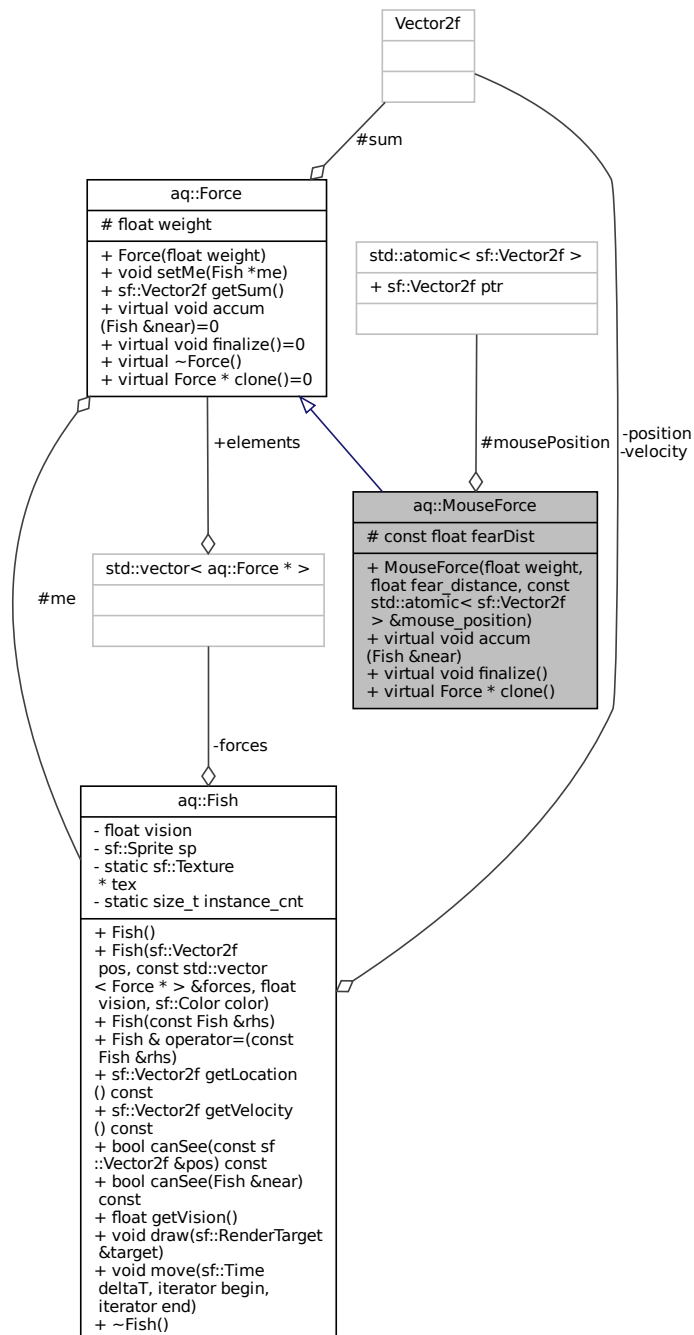
- inc/forces.hpp

3.13 aq::MouseForce Class Reference

Inheritance diagram for aq::MouseForce:



Collaboration diagram for aq::MouseForce:



Public Member Functions

- **MouseForce** (float weight, float fear_distance, const std::atomic< sf::Vector2f > &mouse_position)
- virtual void **accum** (Fish &near)
- virtual void **finalize** ()
- virtual Force * **clone** ()

Protected Attributes

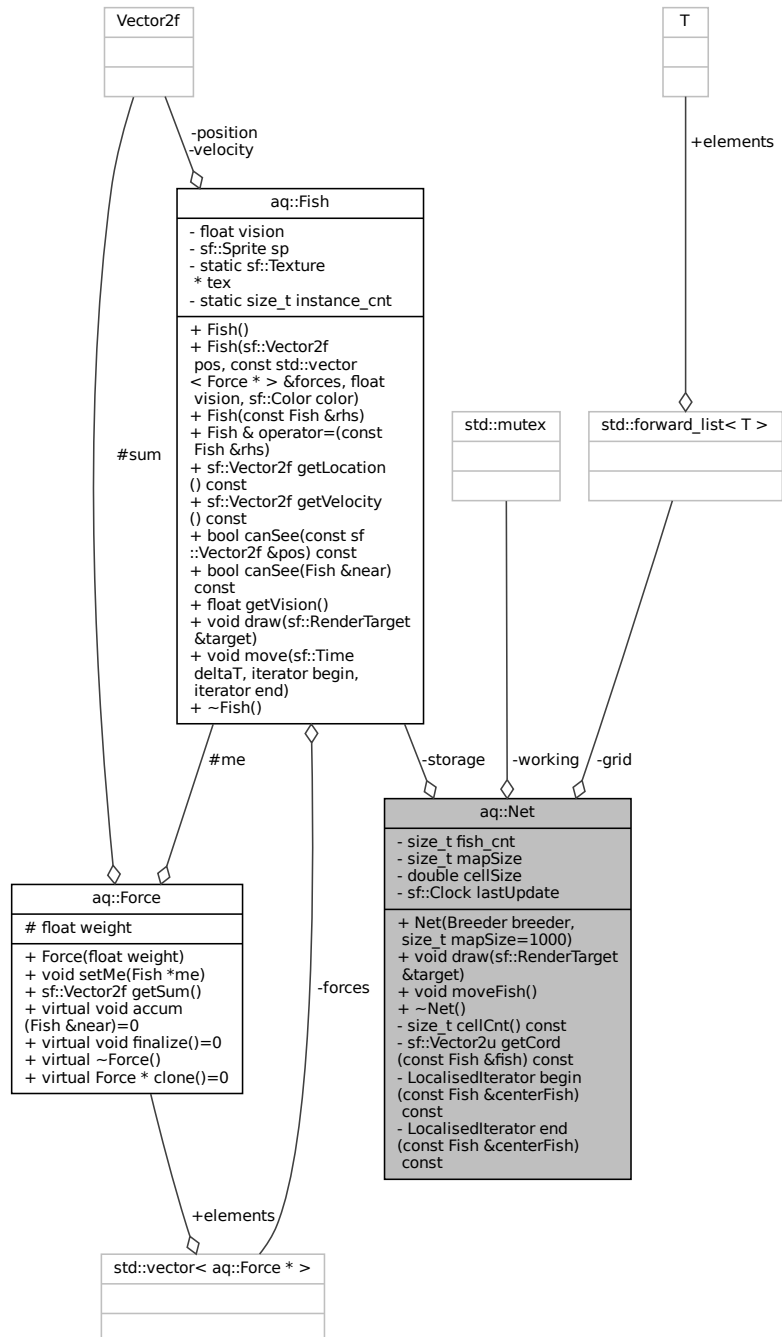
- const float **fearDist**
- const std::atomic< sf::Vector2f > & **mousePosition**

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

- inc/forces.hpp

3.14 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](#) ¢erFish) const
- [LocalisedIterator](#) **end** (const [Fish](#) ¢erFish) 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

3.15 [shader::PerlinNoise](#) Class Reference

Simple 2D perlin noise shader.

Collaboration diagram for shader::PerlinNoise:

shader::PerlinNoise
<ul style="list-style-type: none"> + uniform vec2 u_map_size + uniform float u_edge_ratio + 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 and 8 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) + vec2 slope(vec2 pos) + float edgeCurve(vec2 pos) + 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.
- vec2 [slope](#) (vec2 pos)
- float [edgeCurve](#) (vec2 pos)
- void [main](#) ()
Main function.

Public Attributes

- uniform vec2 [u_map_size](#)
Size of the map.
- uniform float [u_edge_ratio](#)
Point where the edge starts to curve up.
- 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](#)
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.

3.15.1 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

3.15.2 Member Function Documentation

3.15.2.1 colorFromHeight()

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

Computes a color based on the height.

Parameters

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

3.15.2.2 fractalNoise()

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

Computes a fractal sum of perlin noise.

Returns

[0, 1]

3.15.2.3 perlin()

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

2D Perlin noise

Parameters

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

Returns

[-1, 1]

3.15.2.4 randomGradient()

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

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

Returns

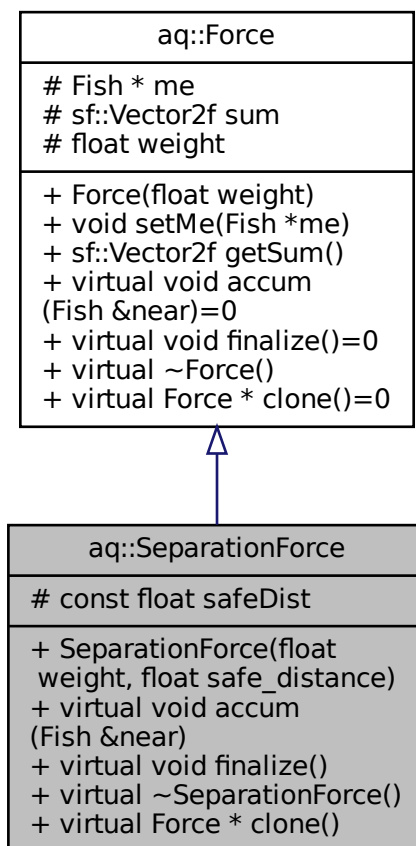
Vector with length 1

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

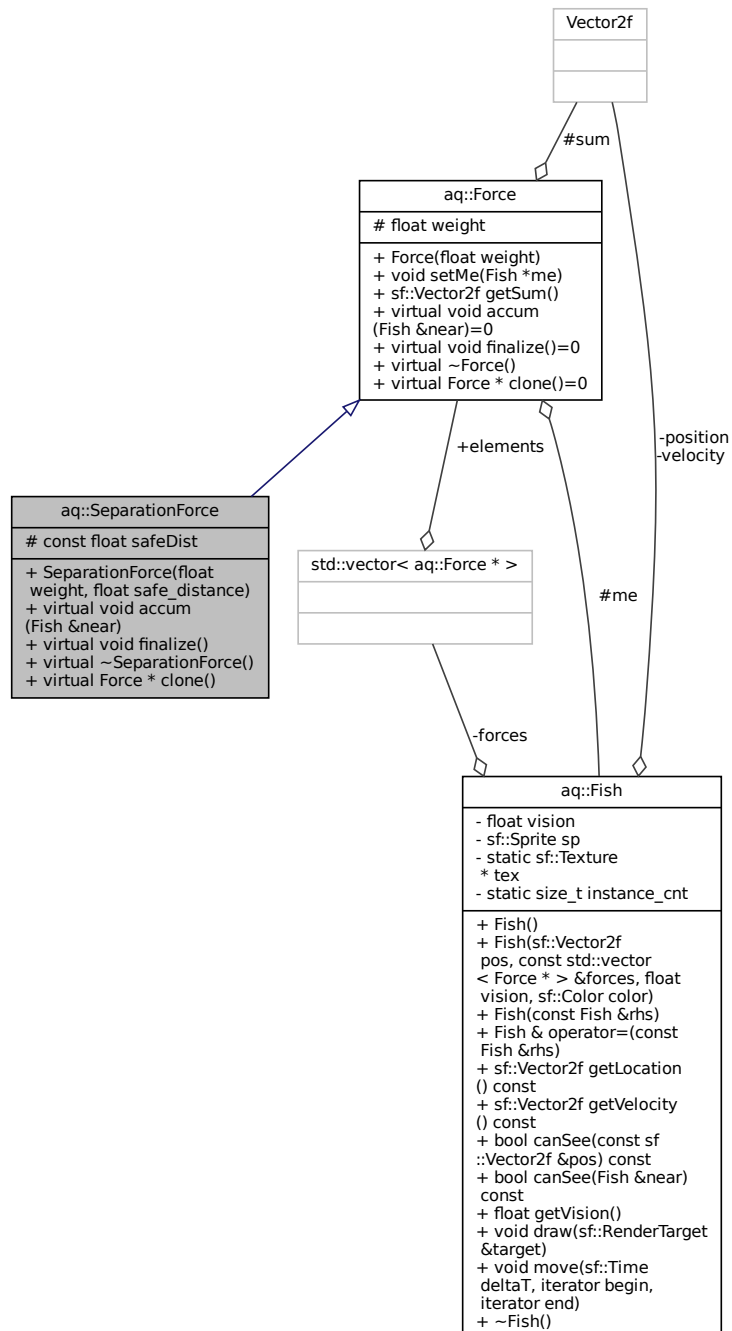
- src/perlin.frag

3.16 aq::SeparationForce Class Reference

Inheritance diagram for aq::SeparationForce:



Collaboration diagram for `aq::SeparationForce`:



Public Member Functions

- **SeparationForce** (float weight, float safe_distance)
- virtual void **accum** ([Fish](#) &near)
- virtual void **finalize** ()
- virtual [Force](#) * **clone** ()

Protected Attributes

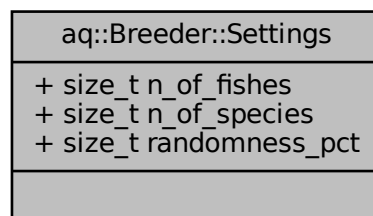
- const float **safeDist**

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

- `inc/forces.hpp`

3.17 `aq::Breeder::Settings` Struct Reference

Collaboration diagram for `aq::Breeder::Settings`:



Public Attributes

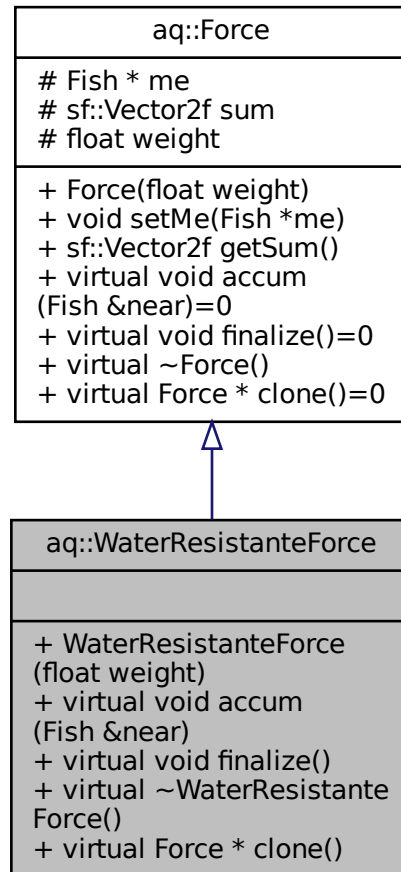
- `size_t n_of_fishes` = 100
- `size_t n_of_species` = 1
- `size_t randomness_pct` = 0

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

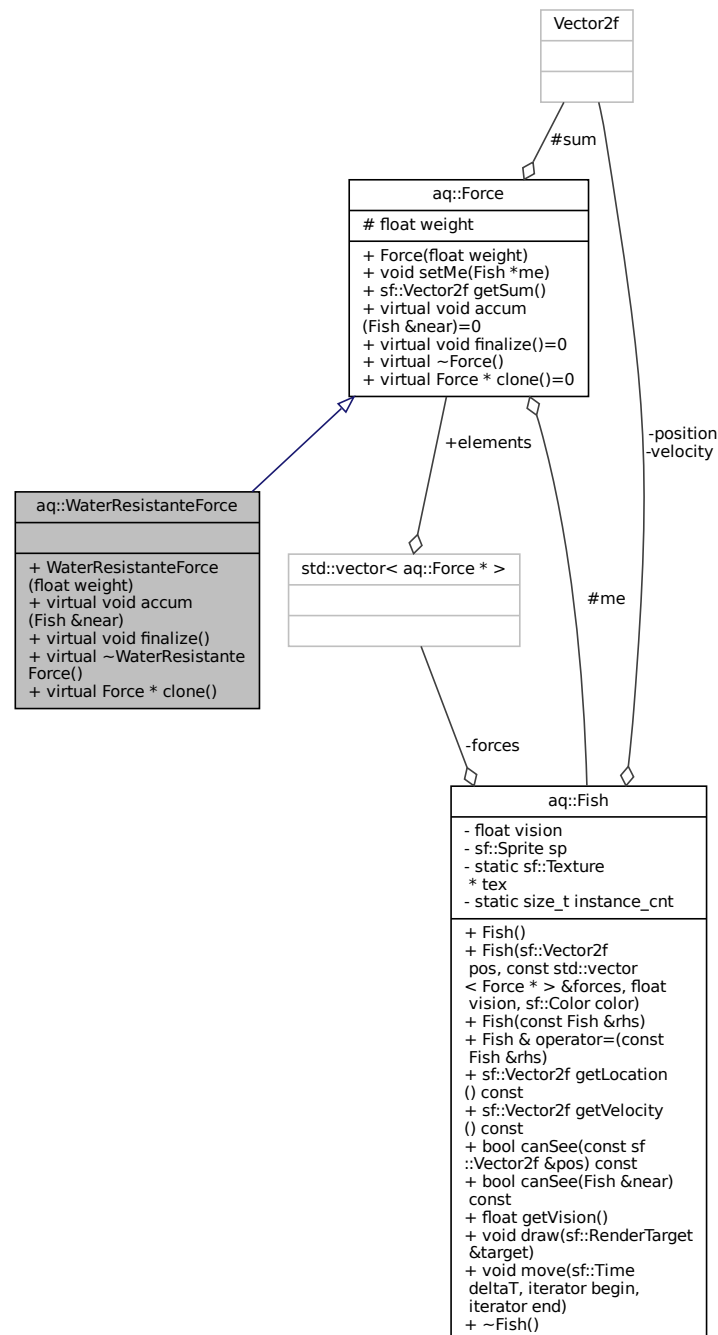
- `inc/breeder.hpp`

3.18 aq::WaterResistenteForce Class Reference

Inheritance diagram for aq::WaterResistenteForce:



Collaboration diagram for aq::WaterResistenteForce:



Public Member Functions

- **WaterResistenteForce** (float weight)
- virtual void **accum** (Fish &near)
- virtual void **finalize** ()
- virtual Force * **clone** ()

Additional Inherited Members

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

- `inc/forces.hpp`

Index

- aq::AlignmentForce, [5](#)
- aq::Breeder, [7](#)
- aq::Breeder::Dependency, [10](#)
- aq::Breeder::Settings, [37](#)
- aq::CohesionForce, [8](#)
- aq::Engine, [11](#)
- aq::Fish, [13](#)
- aq::Force, [14](#)
- aq::Island, [16](#)
- aq::Island::Map, [22](#)
 - operator(), [23](#)
- aq::IslandForce, [18](#)
 - samplePoints, [20](#)
- aq::MinSpeedForce, [24](#)
- aq::MouseForce, [26](#)
- aq::Net, [29](#)
- aq::Net::LocalisedIterator, [21](#)
- aq::SeparationForce, [35](#)
- aq::WaterResistantForce, [38](#)

- colorFromHeight
 - shader::PerlinNoise, [33](#)

- fractalNoise
 - shader::PerlinNoise, [33](#)

- operator()
 - aq::Island::Map, [23](#)

- perlin
 - shader::PerlinNoise, [33](#)

- randomGradient
 - shader::PerlinNoise, [34](#)

- samplePoints
 - aq::IslandForce, [20](#)
- shader::PerlinNoise, [30](#)
 - colorFromHeight, [33](#)
 - fractalNoise, [33](#)
 - perlin, [33](#)
 - randomGradient, [34](#)