Aqua

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::Engine Class Reference

Collaboration diagram for aq::Engine:
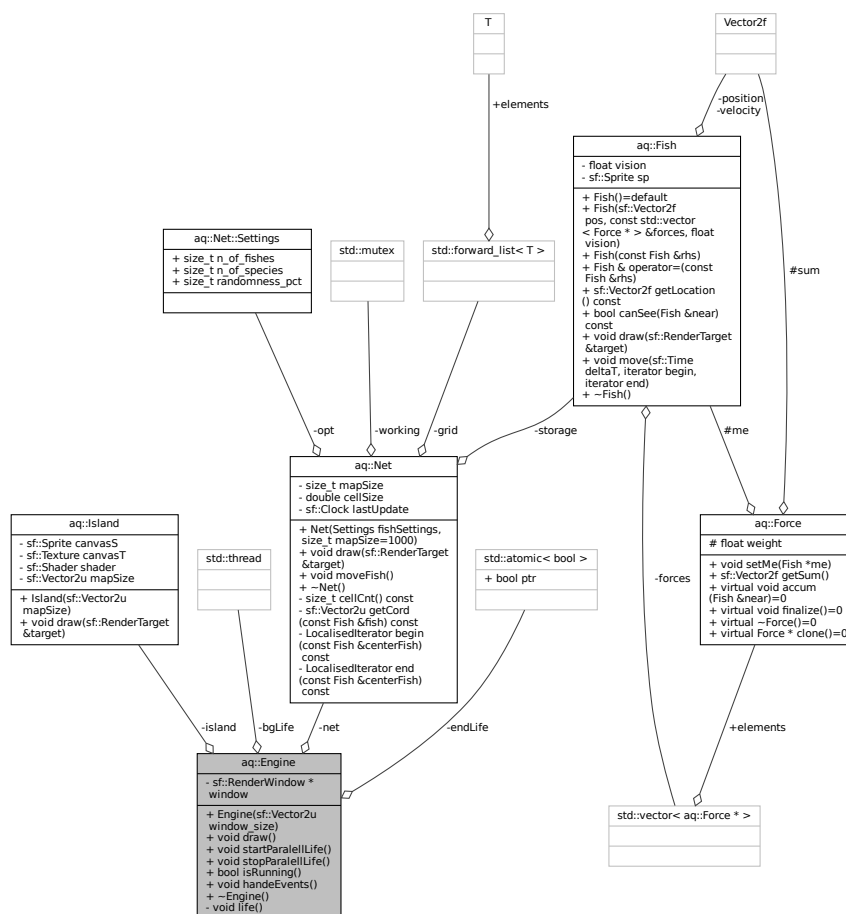
Public Member Functions

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

Private Member Functions

- void **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 &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**
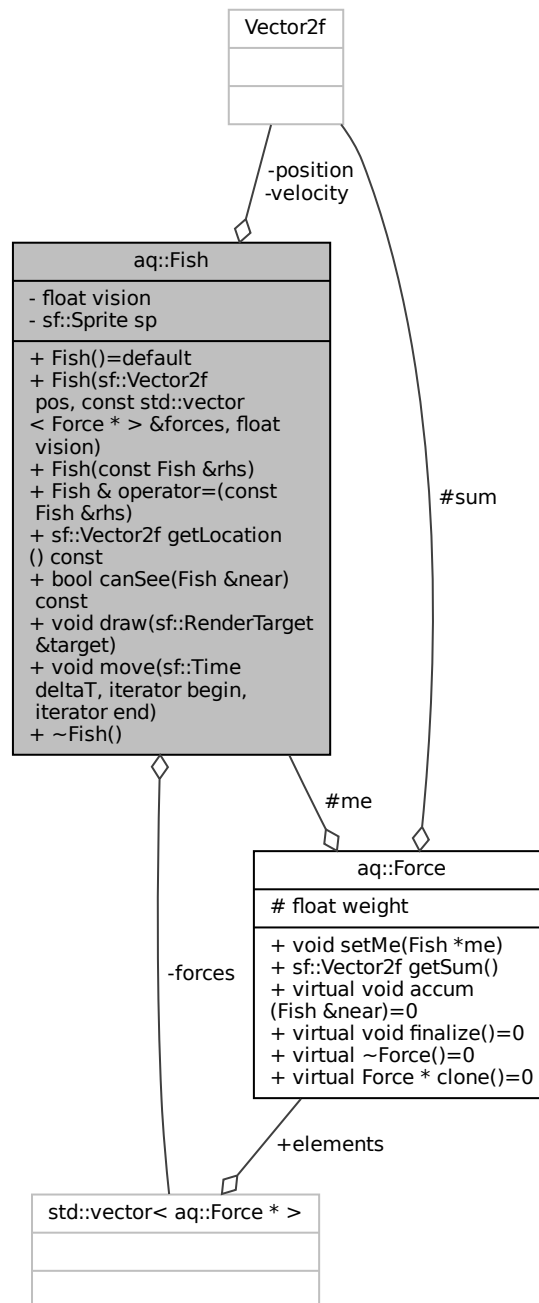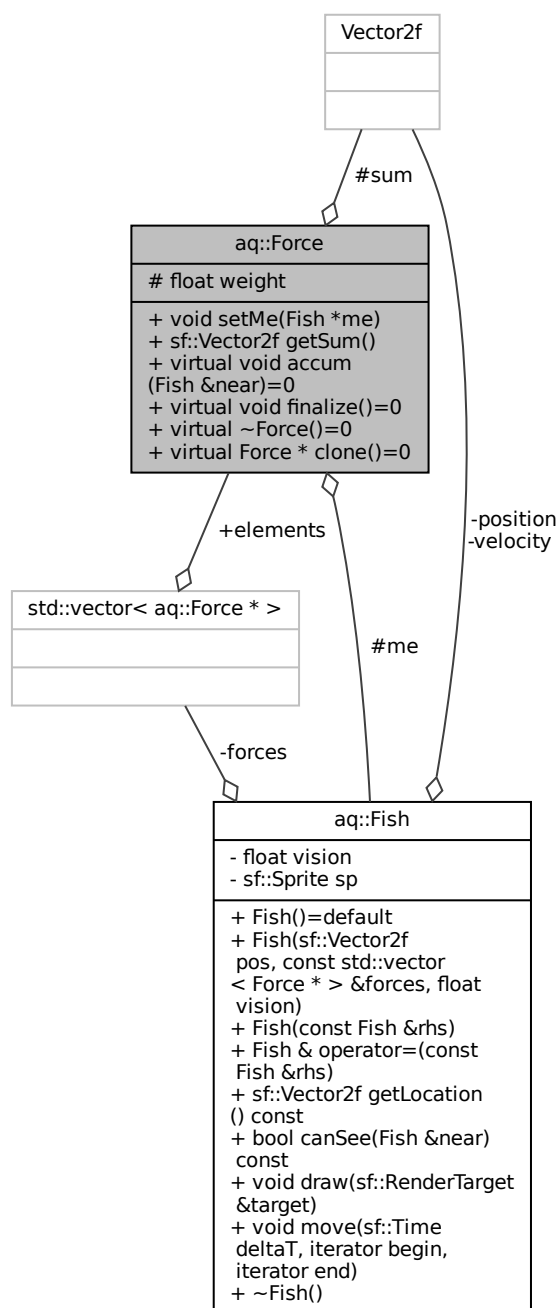- std::vector< Force ∗ > **forces**
- float **vision**
- sf::Sprite **sp**

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:

```
                              ┌──────────────┐
                              │   Vector2f   │
                              ├──────────────┤
                              ├──────────────┤
                              └──────────────┘
                                  │       ╲
                                #sum       ╲
                              ◇             ╲
                    ┌────────────────────┐   ╲
                    │      aq::Force     │    ╲
                    ├────────────────────┤     ╲
                    │ # float weight     │      ╲
                    ├────────────────────┤       ╲
                    │ + void setMe(Fish *me)      ╲
                    │ + sf::Vector2f getSum()      │
                    │ + virtual void accum         │
                    │ (Fish &near)=0        -position
                    │ + virtual void finalize()=0 -velocity
                    │ + virtual ~Force()=0        │
                    │ + virtual Force * clone()=0 │
                    └────────────────────┘        │
                       ╱              ◇           │
                 +elements           #me          │
                   ◇                  │           │
       ┌────────────────────────┐     │           │
       │ std::vector< aq::Force * > │  │           │
       ├────────────────────────┤     │           │
       ├────────────────────────┤     │           │
       └────────────────────────┘     │           │
                   ╲                   │           │
                 -forces               │           │
                   ◇                   │           ◇
              ┌──────────────────────────────┐
              │           aq::Fish           │
              ├──────────────────────────────┤
              │ - float vision               │
              │ - sf::Sprite sp              │
              ├──────────────────────────────┤
              │ + Fish()=default             │
              │ + 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 &near)    │
              │  const                       │
              │ + void draw(sf::RenderTarget │
              │  &target)                    │
              │ + void move(sf::Time         │
              │  deltaT, iterator begin,     │
              │  iterator end)               │
              │ + ~Fish()                    │
              └──────────────────────────────┘
```

Public Member Functions

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

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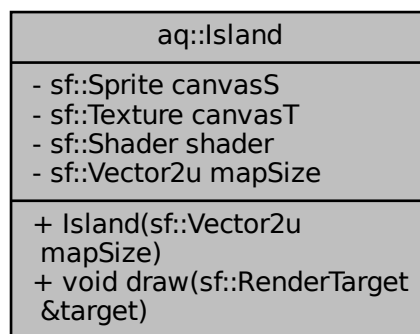- virtual Force ∗ **clone** ()=0

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

2.4 aq::Island Class Reference

Collaboration diagram for aq::Island:

aq::Island
- sf::Sprite canvasS - sf::Texture canvasT - sf::Shader shader - sf::Vector2u mapSize

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

Public Member Functions

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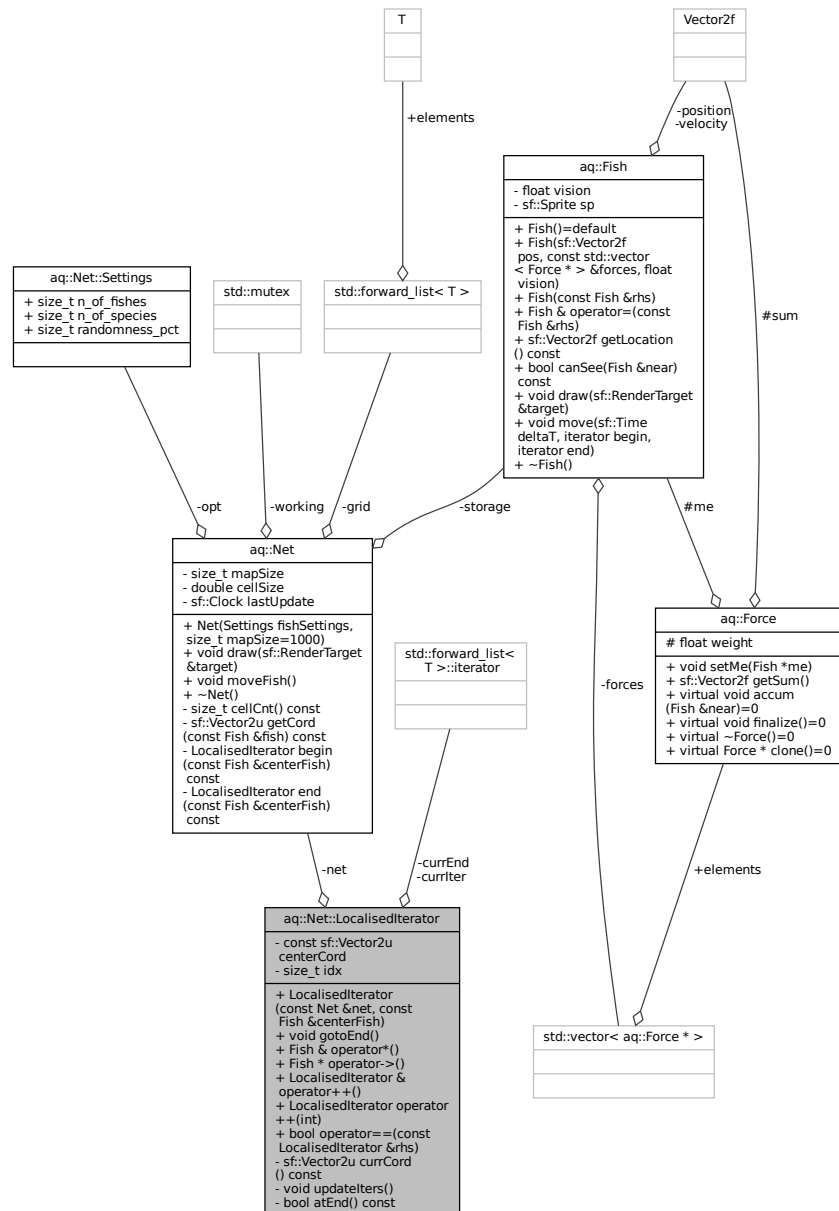
Private Attributes

- 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 &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 file:

- inc/net.hpp

2.6 aq::Net Class Reference

Collaboration diagram for aq::Net:

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

Private Attributes
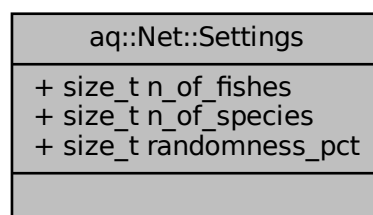
- const Settings **opt**
- Fish ∗ **storage**
- cell ∗∗ **grid**
- size_t **mapSize**
- double **cellSize**
- sf::Clock **lastUpdate**
- std::mutex **working**

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

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

2.7 aq::Net::Settings Struct Reference

Collaboration diagram for aq::Net::Settings:

aq::Net::Settings
+ size_t n_of_fishes
+ size_t n_of_species
+ size_t randomness_pct

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