assign2

Generated by Doxygen 1.8.9.1

Wed Sep 30 2015 12:54:42

Contents

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 Agent Class Reference

```
#include <agent.hpp>
```

Public Member Functions

- **Agent** (**options_t** &options)
- ∼**Agent** (void)
- **lifetime_t lifetime** (void) const
- **reward_t reward** (void) const
- **reward_t averageReward** (void) const
- **reward_t maxReward** (void) const
- **reward_t minReward** (void) const
- unsigned int **numActions** (void) const
- size_t **historySize** (void) const
- size_t **horizon** (void) const
- **action_t genRandomAction** (void) const
- **percept_t genPercept** (void) const
- **percept_t genPerceptAndUpdate** (void)
- void **modelUpdate** (**percept_t** observation, **percept_t reward**)
- void **modelUpdate** (**action_t** action)
- bool **modelRevert** (const **ModelUndo** &mu)
- void **reset** (void)
- double **getPredictedActionProb** (**action_t** action)
- double **perceptProbability** (**percept_t** observation, **percept_t reward**) const

4.1.1 Detailed Description

Definition at line 12 of file agent.hpp.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Agent::Agent (options_t & *options*)

Definition at line 11 of file agent.cpp.

4.1.2.2 Agent::∼Agent (void)

Definition at line 31 of file agent.cpp.

4.1.3 Member Function Documentation

4.1.3.1 reward_t Agent::averageReward (void) const

Definition at line 48 of file agent.cpp.

4.1.3.2 percept_t Agent::genPercept (void) const

Definition at line 89 of file agent.cpp.

4.1.3.3 percept_t Agent::genPerceptAndUpdate (void)

Definition at line 96 of file agent.cpp.

4.1.3.4 action_t Agent::genRandomAction (void) const

Definition at line 83 of file agent.cpp.

4.1.3.5 double Agent::getPredictedActionProb (action_t *action*)

Definition at line 150 of file agent.cpp.

4.1.3.6 size_t Agent::historySize (void) const

Definition at line 71 of file agent.cpp.

4.1.3.7 size_t Agent::horizon (void) const

Definition at line 77 of file agent.cpp.

4.1.3.8 lifetime_t Agent::lifetime (void) const

Definition at line 37 of file agent.cpp.

4.1.3.9 reward_t Agent::maxReward (void) const

Definition at line 53 of file agent.cpp.

4.1.3.10 reward_t Agent::minReward (void) const

Definition at line 59 of file agent.cpp.

4.1.3.11 bool Agent::modelRevert (const ModelUndo & *mu*)

Definition at line 135 of file agent.cpp.

4.1.3.12 void Agent::modelUpdate (percept_t *observation*, percept_t *reward*)

Definition at line 102 of file agent.cpp.

4.1.3.13 void Agent::modelUpdate (action_t *action*)

Definition at line 118 of file agent.cpp.

4.1.3.14 unsigned int Agent::numActions (void) const

Definition at line 65 of file agent.cpp.

4.1.3.15 double Agent::perceptProbability (percept_t *observation*, percept_t *reward*) const

Definition at line 156 of file agent.cpp.

4.1.3.16 void Agent::reset (void)

Definition at line 140 of file agent.cpp.

4.1.3.17 reward_t Agent::reward (void) const

Definition at line 42 of file agent.cpp.

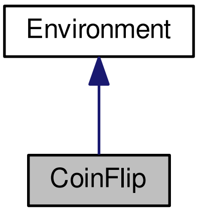
The documentation for this class was generated from the following files:

- **agent.hpp**
- **agent.cpp**

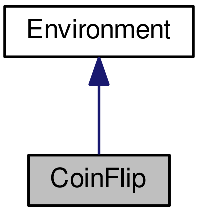
4.2 CoinFlip Class Reference

`#include <environment.hpp>`

Inheritance diagram for CoinFlip:



Collaboration diagram for CoinFlip:



Public Member Functions

- **CoinFlip** (**options_t** &options)
- virtual void **performAction** (**action_t** action)

Additional Inherited Members

4.2.1 Detailed Description

Definition at line 36 of file environment.hpp.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 CoinFlip::CoinFlip (options_t & *options*)

Definition at line 7 of file environment.cpp.

4.2.3 Member Function Documentation

4.2.3.1 void CoinFlip::performAction (action_t *action*) [virtual]

Implements **Environment** (p. 15).

Definition at line 24 of file environment.cpp.

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

- **environment.hpp**
- **environment.cpp**

4.3 ContextTree Class Reference

```
#include <predict.hpp>
```

Public Member Functions

- **ContextTree** (size_t **depth**)
- ∼**ContextTree** (void)
- void **clear** (void)
- void **update** (const **symbol_t** sym)
- void **update** (const **symbol_list_t** &symbol_list)
- void **updateHistory** (const **symbol_list_t** &symbol_list)
- void **updateLogProbability** (void)
- void **revert** (void)
- void **revertHistory** (size_t newsize)
- double **predict** (**symbol_t** sym)
- double **predict** (**symbol_list_t** symbol_list)
- void **genRandomSymbols** (**symbol_list_t** &symbols, size_t bits)
- void **genRandomSymbolsAndUpdate** (**symbol_list_t** &symbols, size_t bits)
- double **logBlockProbability** (void)
- const **symbol_t** ∗ **nthHistorySymbol** (size_t n) const
- size_t **depth** (void) const
- size_t **historySize** (void) const
- size_t **size** (void) const

4.3.1 Detailed Description

Definition at line 69 of file predict.hpp.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 ContextTree::ContextTree (size_t *depth*)

Definition at line 53 of file predict.cpp.

4.3.2.2 ContextTree::∼ContextTree (void)

Definition at line 59 of file predict.cpp.

4.3.3 Member Function Documentation

4.3.3.1 void ContextTree::clear (void)

Definition at line 65 of file predict.cpp.

4.3.3.2 size_t ContextTree::depth (void) const [inline]

Definition at line 116 of file predict.hpp.

4.3.3.3 void ContextTree::genRandomSymbols (symbol_list_t & *symbols,* size_t *bits*)

Definition at line 108 of file predict.cpp.

4.3.3.4 void ContextTree::genRandomSymbolsAndUpdate (symbol_list_t & *symbols,* size_t *bits*)

Definition at line 120 of file predict.cpp.

4.3.3.5 size_t ContextTree::historySize (void) const [inline]

Definition at line 119 of file predict.hpp.

4.3.3.6 double ContextTree::logBlockProbability (void)

Definition at line 126 of file predict.cpp.

4.3.3.7 const symbol_t ∗ ContextTree::nthHistorySymbol (size_t *n*) const

Definition at line 132 of file predict.cpp.

4.3.3.8 double ContextTree::predict (symbol_t *sym*)

4.3.3.9 double ContextTree::predict (symbol_list_t *symbol_list*)

4.3.3.10 void ContextTree::revert (void)

Definition at line 92 of file predict.cpp.

4.3.3.11 void ContextTree::revertHistory (size_t *newsize*)

Definition at line 98 of file predict.cpp.

4.3.3.12 size_t ContextTree::size (void) const [inline]

Definition at line 122 of file predict.hpp.

4.3.3.13 void ContextTree::update (const symbol_t *sym*)

Definition at line 72 of file predict.cpp.

4.3.3.14 void ContextTree::update (const symbol_list_t & *symbol_list*)

Definition at line 77 of file predict.cpp.

4.3.3.15 void ContextTree::updateHistory (const symbol_list_t & *symbol_list*)

Definition at line 83 of file predict.cpp.

4.3.3.16 void ContextTree::updateLogProbability (void)

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

- **predict.hpp**
- **predict.cpp**

4.4 CTNode Class Reference

```
#include <predict.hpp>
```

Public Member Functions

- **weight_t logProbWeighted** (void) const
- **weight_t logProbEstimated** (void) const
- **count_t visits** (void) const
- const **CTNode** ∗ **child** (**symbol_t** sym) const
- size_t **size** (void) const

Friends

- class **ContextTree**

4.4.1 Detailed Description

Definition at line 17 of file predict.hpp.

4.4.2 Member Function Documentation

4.4.2.1 const CTNode∗ CTNode::child (symbol_t *sym*) const [inline]

Definition at line 31 of file predict.hpp.

4.4.2.2 weight_t CTNode::logProbEstimated (void) const [inline]

Definition at line 25 of file predict.hpp.

4.4.2.3 weight_t CTNode::logProbWeighted (void) const [inline]

Definition at line 22 of file predict.hpp.

4.4.2.4 size_t CTNode::size (void) const

Definition at line 24 of file predict.cpp.

4.4.2.5 count_t CTNode::visits (void) const [inline]

Definition at line 28 of file predict.hpp.

4.4.3 Friends And Related Function Documentation

4.4.3.1 friend class ContextTree [friend]

Definition at line 18 of file predict.hpp.

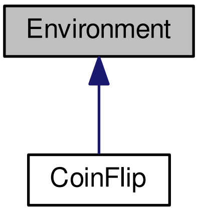
The documentation for this class was generated from the following files:

- **predict.hpp**
- **predict.cpp**

4.5 Environment Class Reference

`#include <environment.hpp>`

Inheritance diagram for Environment:



Public Member Functions

- virtual void **performAction** (**action_t** action)=0
- virtual bool **isFinished** (void) const
- void **getPercept** (**symbol_list_t** &symbol_list)
- **percept_t** **getObservation** (void) const
- **percept_t** **getReward** (void) const

Protected Attributes

- **action_t m_last_action**
- **percept_t m_observation**
- **percept_t m_reward**

4.5.1 Detailed Description

Definition at line 6 of file environment.hpp.

4.5.2 Member Function Documentation

4.5.2.1 percept_t Environment::getObservation (void) const `[inline]`

Definition at line 21 of file environment.hpp.

4.5.2.2 void Environment::getPercept (symbol_list_t & *symbol_list*)

4.5.2.3 percept_t Environment::getReward (void) const `[inline]`

Definition at line 23 of file environment.hpp.

4.5.2.4 virtual bool Environment::isFinished (void) const `[inline], [virtual]`

Definition at line 17 of file environment.hpp.

4.5.2.5 virtual void Environment::performAction (action_t *action*) [pure virtual]

Implemented in **CoinFlip** (p. 10).

4.5.3 Member Data Documentation

4.5.3.1 action_t Environment::m_last_action [protected]

Definition at line 26 of file environment.hpp.

4.5.3.2 percept_t Environment::m_observation [protected]

Definition at line 27 of file environment.hpp.

4.5.3.3 percept_t Environment::m_reward [protected]

Definition at line 28 of file environment.hpp.

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

- **environment.hpp**

4.6 ModelUndo Class Reference

`#include <agent.hpp>`

Public Member Functions

- **ModelUndo** (const **Agent** &agent)
- **lifetime_t lifetime** (void) const
- **reward_t reward** (void) const
- size_t **historySize** (void) const
- bool **lastUpdate** (void) const

4.6.1 Detailed Description

Definition at line 113 of file agent.hpp.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 ModelUndo::ModelUndo (const Agent & *agent*)

Definition at line 201 of file agent.cpp.

4.6.3 Member Function Documentation

4.6.3.1 size_t ModelUndo::historySize (void) const [inline]

Definition at line 126 of file agent.hpp.

4.6.3.2 bool ModelUndo::lastUpdate (void) const [inline]

Definition at line 128 of file agent.hpp.

4.6.3.3 lifetime_t ModelUndo::lifetime (void) const [inline]

Definition at line 120 of file agent.hpp.

4.6.3.4 reward_t ModelUndo::reward (void) const [inline]

Definition at line 123 of file agent.hpp.

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

- **agent.hpp**
- **agent.cpp**

4.7 SearchNode Class Reference

Public Member Functions

- **SearchNode** (bool is_chance_node)
- **action_t selectAction** (**Agent** &agent) const
- **reward_t expectation** (void) const
- **reward_t sample** (**Agent** &agent, unsigned int dfr)
- **visits_t visits** (void) const

4.7.1 Detailed Description

Definition at line 14 of file search.cpp.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 SearchNode::SearchNode (bool *is_chance_node*)

4.7.3 Member Function Documentation

4.7.3.1 reward_t SearchNode::expectation (void) const [inline]

Definition at line 24 of file search.cpp.

4.7.3.2 reward_t SearchNode::sample (Agent & *agent,* unsigned int *dfr*)

4.7.3.3 action_t SearchNode::selectAction (Agent & *agent*) const

4.7.3.4 visits_t SearchNode::visits (void) const [inline]

Definition at line 31 of file search.cpp.

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

- **search.cpp**

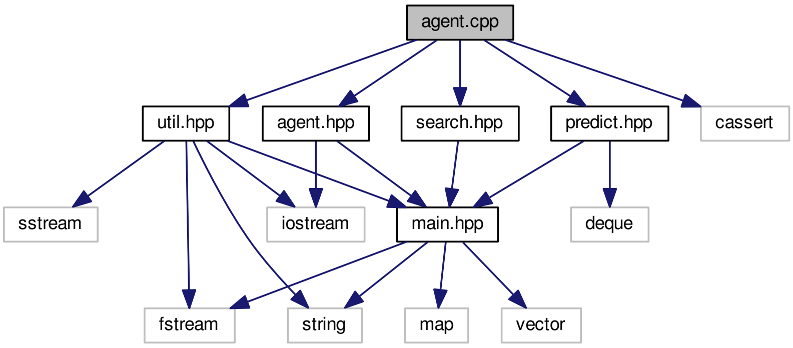
Chapter 5

File Documentation

5.1 agent.cpp File Reference

```
#include "agent.hpp"
#include <cassert>
#include "predict.hpp"
#include "search.hpp"
#include "util.hpp"
```

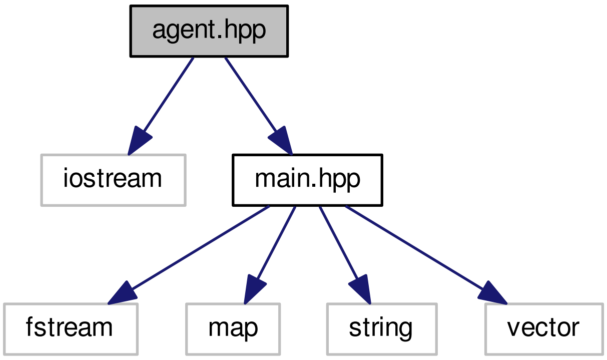
Include dependency graph for agent.cpp:



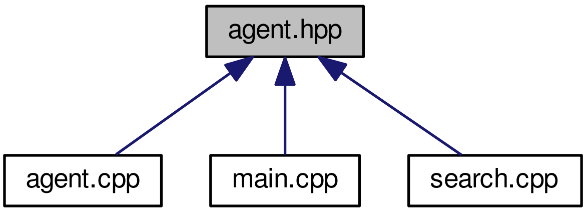
5.2 agent.hpp File Reference

```
#include <iostream>
#include "main.hpp"
```

Include dependency graph for agent.hpp:



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



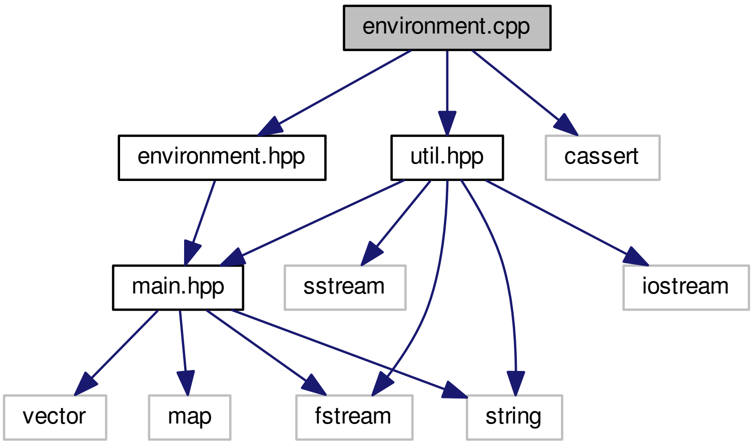
Classes

- class **Agent**
- class **ModelUndo**

5.3 environment.cpp File Reference

```
#include "environment.hpp"
#include <cassert>
#include "util.hpp"
```

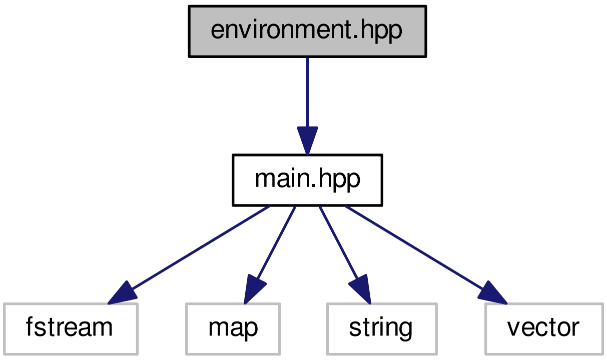
Include dependency graph for environment.cpp:



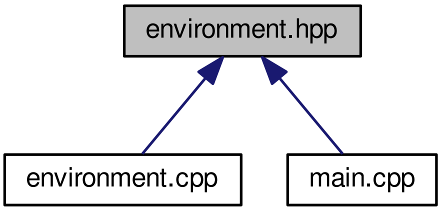
5.4 environment.hpp File Reference

```
#include "main.hpp"
```

Include dependency graph for environment.hpp:



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



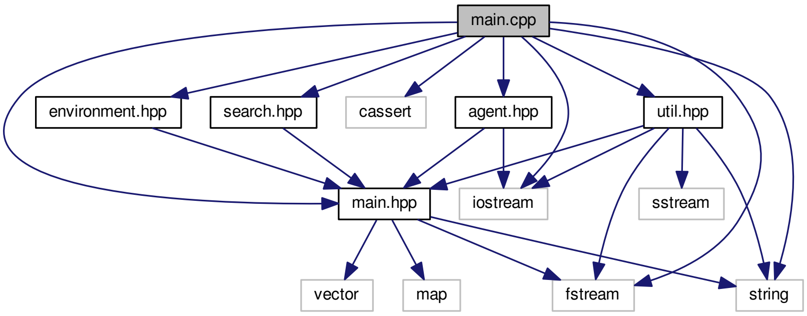
Classes

- class **Environment**
- class **CoinFlip**

5.5 main.cpp File Reference

```
#include "main.hpp"
#include <cassert>
#include <fstream>
#include <iostream>
#include <string>
#include "agent.hpp"
#include "environment.hpp"
#include "search.hpp"
#include "util.hpp"
```

Include dependency graph for main.cpp:



Functions

- void **mainLoop** (**Agent** &ai, **Environment** &env, **options_t** &options)
- void **processOptions** (std::ifstream &in, **options_t** &options)
- int **main** (int argc, char ∗argv[ ])

Variables

- std::ofstream **log**
- std::ofstream **compactLog**

5.5.1 Function Documentation

5.5.1.1 int main (int *argc*, char ∗ *argv[ ]*)

Definition at line 153 of file main.cpp.

5.5.1.2 void mainLoop (Agent & *ai*, Environment & *env*, options_t & *options*)

Definition at line 19 of file main.cpp.

5.5.1.3 void processOptions (std::ifstream & *in*, options_t & *options*)

Definition at line 110 of file main.cpp.

5.5.2 Variable Documentation

5.5.2.1 std::ofstream compactLog

Definition at line 16 of file main.cpp.

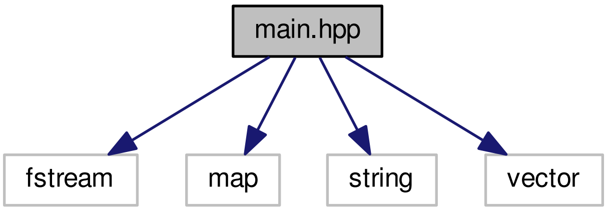
5.5.2.2 std::ofstream log

Definition at line 15 of file main.cpp.

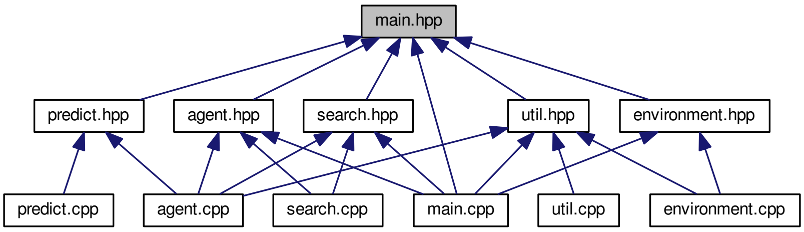
5.6 main.hpp File Reference

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

Include dependency graph for main.hpp:



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



Typedefs

- typedef bool **symbol_t**
- typedef std::vector< **symbol_t** > **symbol_list_t**
- typedef double **reward_t**
- typedef unsigned int **percept_t**
- typedef unsigned long long **lifetime_t**
- typedef unsigned int **action_t**
- typedef std::map< std::string, std::string > **options_t**

Variables

- std::ofstream **log**
- std::ofstream **compactLog**

5.6.1 Typedef Documentation

5.6.1.1 typedef unsigned int **action_t**

Definition at line 29 of file main.hpp.

5.6.1.2 typedef unsigned long long **lifetime_t**

Definition at line 26 of file main.hpp.

5.6.1.3 typedef std::map<std::string, std::string> **options_t**

Definition at line 32 of file main.hpp.

5.6.1.4 typedef unsigned int **percept_t**

Definition at line 23 of file main.hpp.

5.6.1.5 typedef double **reward_t**

Definition at line 20 of file main.hpp.

5.6.1.6 typedef std::vector<symbol_t> symbol_list_t

Definition at line 17 of file main.hpp.

5.6.1.7 typedef bool symbol_t

Definition at line 14 of file main.hpp.

5.6.2 Variable Documentation

5.6.2.1 std::ofstream compactLog

Definition at line 16 of file main.cpp.

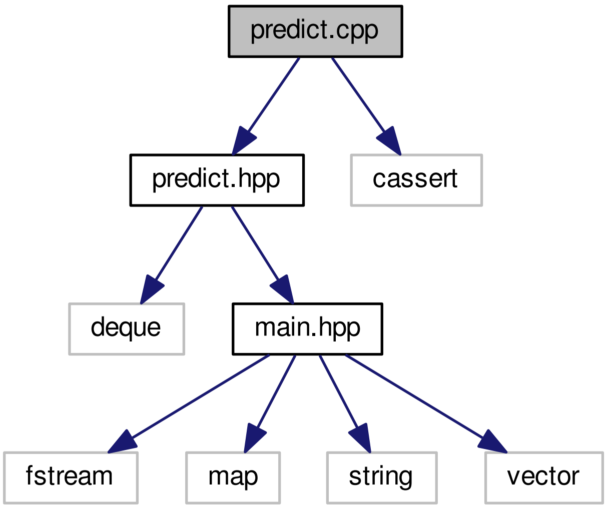
5.6.2.2 std::ofstream log

Definition at line 15 of file main.cpp.

5.7 predict.cpp File Reference

```
#include "predict.hpp"
#include <cassert>
```

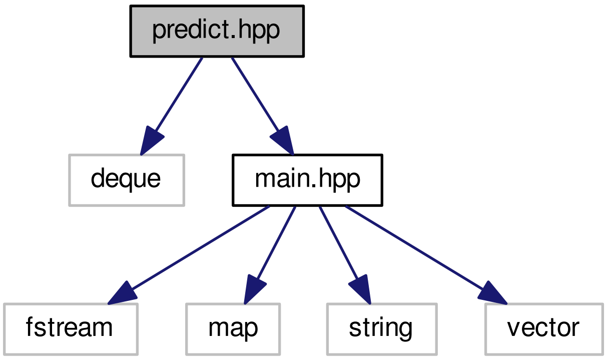
Include dependency graph for predict.cpp:



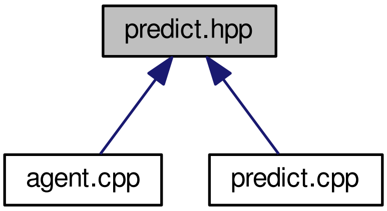
5.8 predict.hpp File Reference

```
#include <deque>
#include "main.hpp"
```

Include dependency graph for predict.hpp:



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



Classes

- class **CTNode**
- class **ContextTree**

Typedefs

- typedef unsigned int **count_t**
- typedef double **weight_t**
- typedef std::deque< **symbol_t** > **history_t**

5.8.1 Typedef Documentation

5.8.1.1 typedef unsigned int count_t

Definition at line 9 of file predict.hpp.

5.8.1.2 typedef std::deque<symbol_t> history_t

Definition at line 15 of file predict.hpp.

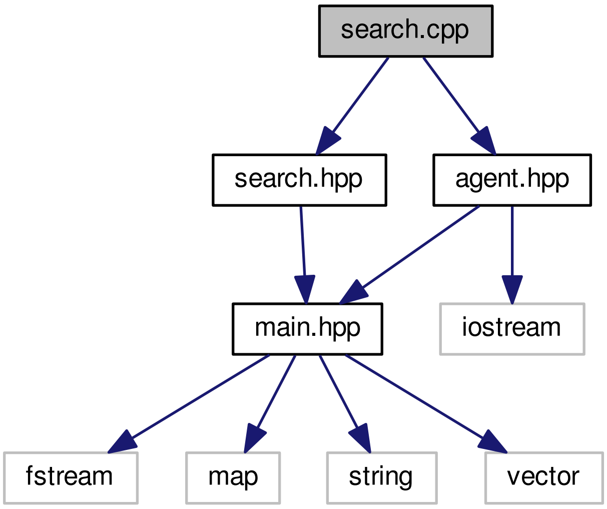
5.8.1.3 typedef double weight_t

Definition at line 12 of file predict.hpp.

5.9 search.cpp File Reference

```
#include "search.hpp"
#include "agent.hpp"
```

Include dependency graph for search.cpp:



Classes

- class **SearchNode**

Typedefs

- typedef unsigned long long **visits_t**

Functions

- **action_t search** (**Agent** &agent)

5.9.1 Typedef Documentation

5.9.1.1 typedef unsigned long long visits_t

Definition at line 5 of file search.cpp.

5.9.2 Function Documentation

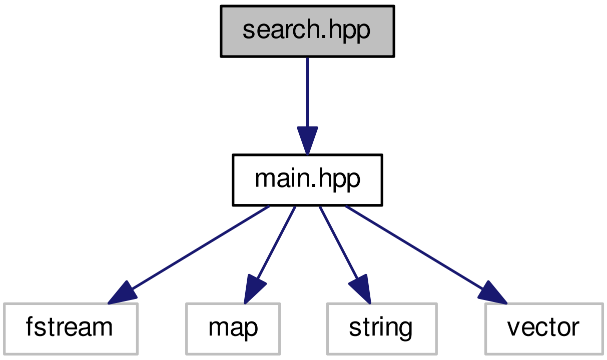
5.9.2.1 action_t search (Agent & *agent*)

Definition at line 50 of file search.cpp.

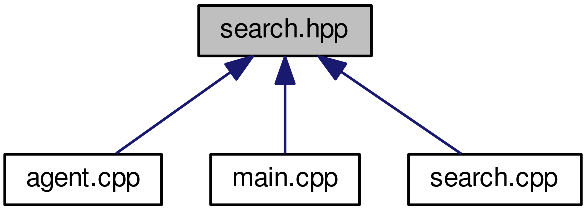
5.10 search.hpp File Reference

```
#include "main.hpp"
```

Include dependency graph for search.hpp:



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



Functions

- **action_t search** (**Agent** &agent)

5.10.1 Function Documentation

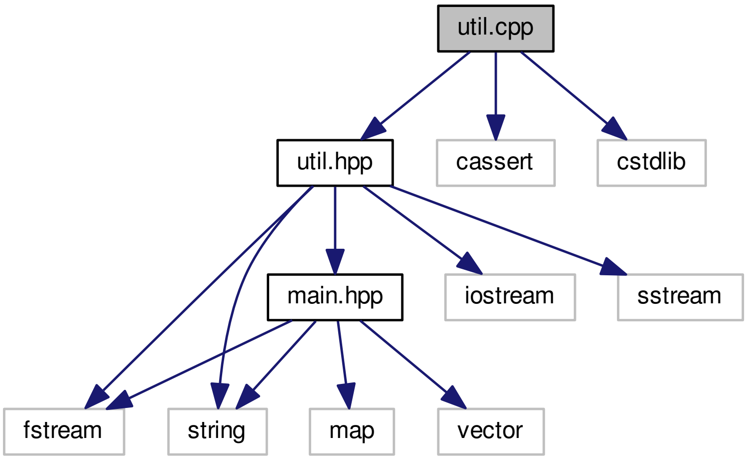
5.10.1.1 action_t search (Agent & *agent*)

Definition at line 50 of file search.cpp.

5.11 util.cpp File Reference

```
#include "util.hpp"
#include <cassert>
#include <cstdlib>
```

Include dependency graph for util.cpp:



Functions

- double **rand01** ()
- unsigned int **randRange** (unsigned int end)
- int **randRange** (int start, int end)
- unsigned int **decode** (const **symbol_list_t** &symlist, unsigned int bits)
- void **encode** (**symbol_list_t** &symlist, unsigned int value, unsigned int bits)

5.11.1 Function Documentation

5.11.1.1 unsigned int decode (const symbol_list_t & *symlist,* unsigned int *bits*)

Definition at line 31 of file util.cpp.

5.11.1.2 void encode (symbol_list_t & *symlist*, unsigned int *value*, unsigned int *bits*)

Definition at line 46 of file util.cpp.

5.11.1.3 double rand01 ( )

Definition at line 8 of file util.cpp.

5.11.1.4 unsigned int randRange (unsigned int *end*)

Definition at line 13 of file util.cpp.

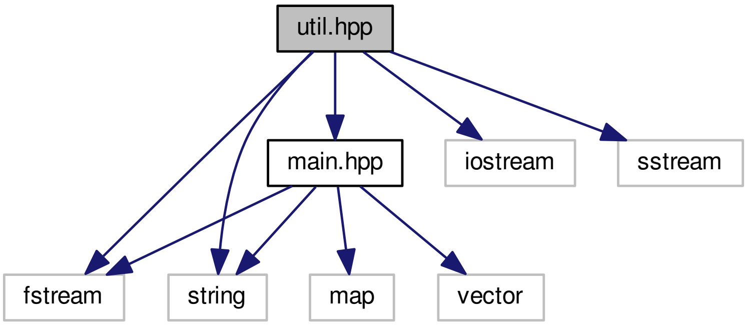
5.11.1.5 int randRange (int *start*, int *end*)

Definition at line 24 of file util.cpp.

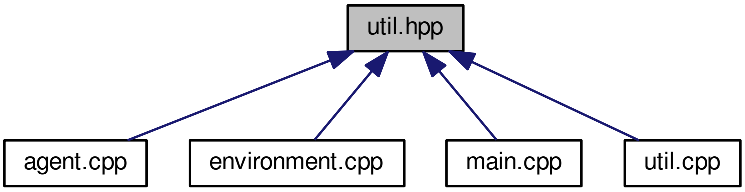
5.12 util.hpp File Reference

```
#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include "main.hpp"
```

Include dependency graph for util.hpp:



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



Functions

- double **rand01** ()
- unsigned int **randRange** (unsigned int end)
- int **randRange** (int start, int end)
- template<typename T >
  void **strExtract** (std::string &str, T &val)
- template<typename T >
  T **strExtract** (std::string &str)
- unsigned int **decode** (const **symbol_list_t** &symlist, unsigned int bits)
- void **encode** (**symbol_list_t** &symlist, unsigned int value, unsigned int bits)

5.12.1 Function Documentation

5.12.1.1 unsigned int decode (const symbol_list_t & *symlist*, unsigned int *bits*)

Definition at line 31 of file util.cpp.

5.12.1.2 void encode (symbol_list_t & *symlist*, unsigned int *value*, unsigned int *bits*)

Definition at line 46 of file util.cpp.

5.12.1.3 double rand01 ( )

Definition at line 8 of file util.cpp.

5.12.1.4 unsigned int randRange (unsigned int *end*)

Definition at line 13 of file util.cpp.

5.12.1.5 int randRange (int *start*, int *end*)

Definition at line 24 of file util.cpp.

5.12.1.6 template<typename T > void strExtract (std::string & *str*, T & *val*)

Definition at line 22 of file util.hpp.

5.12.1.7 template$<$typename T $>$ T strExtract (std::string & *str*)

Definition at line 28 of file util.hpp.