

Uniform Search AI

Generated by Doxygen 1.8.18

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 Node Class Reference	5
3.1.1 Detailed Description	5
3.1.2 Member Function Documentation	5
3.1.2.1 operator<()	5
3.1.3 Member Data Documentation	6
3.1.3.1 destination	6
3.1.3.2 dist	6
3.1.3.3 name	6
3.1.3.4 pathDist	6
3.1.3.5 source	6
4 File Documentation	7
4.1 /home/christian/Documents/AI/Temp/CMakeLists.txt File Reference	7
4.2 /home/christian/Documents/AI/Temp/main.cpp File Reference	7
4.2.1 Function Documentation	7
4.2.1.1 main()	8
4.2.2 Variable Documentation	8
4.2.2.1 cities	8
4.2.2.2 connections	8
4.2.2.3 expandedCities	8
4.3 /home/christian/Documents/AI/Temp/node.h File Reference	8
4.4 /home/christian/Documents/AI/Temp/readme.txt File Reference	9
4.4.1 Function Documentation	9
4.4.1.1 string()	9
4.4.2 Variable Documentation	9
4.4.2.1 else	9
4.4.2.2 members	9
4.4.2.3 SECOND	9
Index	11

Chapter 1

Class Index

1.1 Class List

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

Node	A node class to represent a connection in the graph (edge) poorly named	5
----------------------	---	-------------------

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

/home/christian/Documents/AI/Temp/ main.cpp	7
/home/christian/Documents/AI/Temp/ node.h	8

Chapter 3

Class Documentation

3.1 Node Class Reference

A node class to represent a connection in the graph (edge) poorly named.

```
#include <node.h>
```

Public Member Functions

- bool `operator<` (const `Node` &n1)

Public Attributes

- string `name`
- string `destination`
- `Node` * `source`
- int `dist`
- int `pathDist`

3.1.1 Detailed Description

A node class to represent a connection in the graph (edge) poorly named.

3.1.2 Member Function Documentation

3.1.2.1 `operator<()`

```
bool Node::operator< (  
    const Node & n1 ) [inline]
```

3.1.3 Member Data Documentation

3.1.3.1 destination

`string` Node::destination

3.1.3.2 dist

`int` Node::dist

3.1.3.3 name

`string` Node::name

3.1.3.4 pathDist

`int` Node::pathDist

3.1.3.5 source

`Node*` Node::source

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

- `/home/christian/Documents/AI/Temp/node.h`

Chapter 4

File Documentation

4.1 /home/christian/Documents/AI/Temp/CMakeLists.txt File Reference

4.2 /home/christian/Documents/AI/Temp/main.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <queue>
#include <sstream>
#include <stack>
#include "node.h"
```

Functions

- int [main](#) (int argc, char *argv[])

Variables

- vector< [Node](#) * > [connections](#)
- vector< [string](#) > [cities](#)
- vector< [string](#) > [expandedCities](#)

4.2.1 Function Documentation

4.2.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

This is the que ranked of distance from root node

This is for parsing the file

Returns

istringstream

Loop through all connections looking for any that use the origin [Node](#) and add them to the fringeList

Found a path

4.2.2 Variable Documentation

4.2.2.1 cities

```
vector<string> cities
```

4.2.2.2 connections

```
vector<Node *> connections
```

4.2.2.3 expandedCities

```
vector<string> expandedCities
```

4.3 /home/christian/Documents/AI/Temp/node.h File Reference

```
#include <vector>
#include <string>
```

Classes

- class `Node`

A node class to represent a connection in the graph (edge) poorly named.

4.4 /home/christian/Documents/Al/Temp/readme.txt File Reference

Functions

- Group it will return a message that the file was not found and exit The starting city will be saved in a string named `originCity` the final city will be saved in a string named `destinationCity` and the file name will be saved in a string called `inputFile` `string` (`city2`) and `int(theDist)` which is the distance between the two cities. - Each of `city1`

Variables

- Group `members`
- Group `else`
- Group it will return a message that the file was not found and exit The starting city will be saved in a `string` named `originCity` the final city will be saved in a `string` named `destinationCity` and the file name will be saved in a `string` called `inputFile` *** `SECOND`

4.4.1 Function Documentation

4.4.1.1 `string()`

Group it will return a message that the file was not found and exit The starting city will be saved in a string named `originCity` the final city will be saved in a string named `destinationCity` and the file name will be saved in a string called `inputFile` `string` (`city2`)

4.4.2 Variable Documentation

4.4.2.1 `else`

Group `else`

4.4.2.2 `members`

Group `members`

4.4.2.3 `SECOND`

Group it will return a message that the file was not found and exit The starting city will be saved in a `string` named `originCity` the final city will be saved in a `string` named `destinationCity` and the file name will be saved in a `string` called `inputFile`*** `SECOND`

Index

[/home/christian/Documents/AI/Temp/CMakeLists.txt](#), [7](#)
[/home/christian/Documents/AI/Temp/main.cpp](#), [7](#)
[/home/christian/Documents/AI/Temp/node.h](#), [8](#)
[/home/christian/Documents/AI/Temp/readme.txt](#), [9](#)

cities
 [main.cpp](#), [8](#)

connections
 [main.cpp](#), [8](#)

destination
 [Node](#), [6](#)

dist
 [Node](#), [6](#)

else
 [readme.txt](#), [9](#)

expandedCities
 [main.cpp](#), [8](#)

main
 [main.cpp](#), [7](#)

[main.cpp](#)
 [cities](#), [8](#)
 [connections](#), [8](#)
 [expandedCities](#), [8](#)
 [main](#), [7](#)

members
 [readme.txt](#), [9](#)

name
 [Node](#), [6](#)

[Node](#), [5](#)
 [destination](#), [6](#)
 [dist](#), [6](#)
 [name](#), [6](#)
 [operator<](#), [5](#)
 [pathDist](#), [6](#)
 [source](#), [6](#)

[operator<](#)
 [Node](#), [5](#)

[pathDist](#)
 [Node](#), [6](#)

[readme.txt](#)
 [else](#), [9](#)
 [members](#), [9](#)
 [SECOND](#), [9](#)
 [string](#), [9](#)

[SECOND](#)
 [readme.txt](#), [9](#)

[source](#)
 [Node](#), [6](#)

[string](#)
 [readme.txt](#), [9](#)