High Map

0.2.0

Generated by Doxygen 1.9.8

1 Class Index **1**

1.1 Class List . . . 1

2 File Index **3**

2.1 File List . . . 3

3 Class Documentation **5**

3.1 Cube Class Reference . . . 5

3.1.1 Detailed Description . . . 5

3.1.2 Constructor & Destructor Documentation . . . 6

3.1.2.1 Cube() . . . 6

3.1.3 Member Function Documentation . . . 6

3.1.3.1 addNeighbor() . . . 6

3.1.3.2 getNeighborsAtAxis() . . . 6

3.1.3.3 getStatus() . . . 7

3.1.3.4 getX() . . . 7

3.1.3.5 getY() . . . 7

3.1.3.6 getZ() . . . 7

3.1.3.7 setStatus() . . . 7

3.1.3.8 toString() . . . 8

3.1.3.9 walkThroughNeighbors() . . . 8

3.2 CubeScanData Struct Reference . . . 8

3.2.1 Detailed Description . . . 9

3.2.2 Member Data Documentation . . . 9

3.2.2.1 checkIfFaceHasNeighbor . . . 9

3.2.2.2 neighborsAtFaces . . . 9

3.3 HeightMap Class Reference . . . 9

3.3.1 Detailed Description . . . 9

3.3.2 Constructor & Destructor Documentation . . . 9

3.3.2.1 HeightMap() . . . 9

3.3.2.2 ∼HeightMap() . . . 10

3.3.3 Member Function Documentation . . . 10

3.3.3.1 getFaces() . . . 10

4 File Documentation **11**

4.1 Cube.cpp File Reference . . . 11

4.2 Cube.hpp File Reference . . . 11

4.3 Cube.hpp . . . 11

4.4 HeightMap.cpp File Reference . . . 12

4.5 HeightMap.hpp File Reference . . . 12

4.6 HeightMap.hpp . . . 12

4.7 main.cpp File Reference . . . 13

4.7.1 Function Documentation . . . 13

4.7.1.1 main() 13
4.8 types.cpp File Reference 13
4.8.1 Function Documentation 13
4.8.1.1 getAllAxies() 13
4.8.1.2 getCubeDataAtAxis() 14
4.8.1.3 getFacesAtAxis() 15
4.9 types.hpp File Reference 15
4.9.1 Enumeration Type Documentation 16
4.9.1.1 Axis 16
4.9.1.2 Face 16
4.9.1.3 Status 16
4.9.2 Function Documentation 17
4.9.2.1 getAllAxies() 17
4.9.2.2 getCubeDataAtAxis() 17
4.9.2.3 getFacesAtAxis() 17
4.10 types.hpp 17

Index **19**

Chapter 1

Class Index

1.1 Class List

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

Cube
A class that represents a cube 5
CubeScanData
A struct that represents the data of the scanning of a cube from specific axis 8
HeightMap
A class that represents a height map 9

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Cube.cpp 11
Cube.hpp 11
HeightMap.cpp 12
HeightMap.hpp 12
main.cpp 13
types.cpp 13
types.hpp 15

Chapter 3

Class Documentation

3.1 Cube Class Reference

A class that represents a cube.

```
#include <Cube.hpp>
```

Public Member Functions

- Cube (int x, int y, int z)

  Constructs a new Cube object.
- void addNeighbor (Cube *cube, Face face)

  Adds a neighboring cube to the current cube at the specified face.
- int walkThroughNeighbors (Axis axis)

  Walks through the neighbors of the cube along the specified axis.
- std::vector< Cube * > getNeighborsAtAxis (Axis axis)
- void setStatus (Status status)

  Sets the status of the cube.
- int getX () const

  Gets the value of the X coordinate.
- int getY () const

  Gets the Y coordinate of the cube.
- int getZ () const

  Get the Z coordinate of the cube.
- Status getStatus () const

  Get the status of the cube.
- std::string toString () const

  Converts the Cube object to a string representation.

3.1.1 Detailed Description

A class that represents a cube.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Cube()

```
Cube::Cube (
            int x,
            int y,
            int z )
```

Constructs a new Cube object.

Parameters

x	The x-coordinate of the cube.
y	The y-coordinate of the cube.
z	The z-coordinate of the cube.

Returns

A pointer to the newly created Cube object.

3.1.3 Member Function Documentation

3.1.3.1 addNeighbor()

```
void Cube::addNeighbor (
            Cube * cube,
            Face face )
```

Adds a neighboring cube to the current cube at the specified face.

Parameters

cube	Pointer to the neighboring cube.
face	The face of the current cube where the neighboring cube is added.

3.1.3.2 getNeighborsAtAxis()

```
std::vector< Cube * > Cube::getNeighborsAtAxis (
            Axis axis )
```

Retrieves the neighbors of the cube at the specified axis.

Parameters

axis	The axis along which to retrieve the neighbors.

Returns

A vector containing pointers to the neighboring cubes.

3.1.3.3 getStatus()

```
Status Cube::getStatus ( ) const
```

Get the status of the cube.

Returns

The status of the cube.

3.1.3.4 getX()

```
int Cube::getX ( ) const
```

Gets the value of the X coordinate.

Returns

The value of the X coordinate.

3.1.3.5 getY()

```
int Cube::getY ( ) const
```

Gets the Y coordinate of the cube.

Returns

The Y coordinate of the cube.

3.1.3.6 getZ()

```
int Cube::getZ ( ) const
```

Get the Z coordinate of the cube.

Returns

int The Z coordinate of the cube.

3.1.3.7 setStatus()

```
void Cube::setStatus (
             Status status )
```

Sets the status of the cube.

Parameters

status	The new status of the cube.

3.1.3.8 toString()

```
std::string Cube::toString ( ) const
```

Converts the Cube object to a string representation.

Returns

The string representation of the Cube object.

3.1.3.9 walkThroughNeighbors()

```
int Cube::walkThroughNeighbors (
             Axis axis )
```

Walks through the neighbors of the cube along the specified axis.

Parameters

axis	The axis along which to walk through the neighbors.

Returns

1 if the cube has empty face, 0 otherwise.

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

- Cube.hpp
- Cube.cpp

3.2 CubeScanData Struct Reference

A struct that represents the data of the scanning of a cube from specific axis.

```
#include <types.hpp>
```

Public Attributes

- std::vector< Face > neighborsAtFaces

  The faces which we want to look for neighbors.
- Face checkIfFaceHasNeighbor

  The face which must not have neighbor if the cube want to be counted as part of face.

3.2.1 Detailed Description

A struct that represents the data of the scanning of a cube from specific axis.

3.2.2 Member Data Documentation

3.2.2.1 checkIfFaceHasNeighbor

```
Face CubeScanData::checkIfFaceHasNeighbor
```

The face which must not have neighbor if the cube want to be counted as part of face.

3.2.2.2 neighborsAtFaces

```
std::vector<Face> CubeScanData::neighborsAtFaces
```

The faces which we want to look for neighbors.

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

- types.hpp

3.3 HeightMap Class Reference

A class that represents a height map.

```
#include <HeightMap.hpp>
```

Public Member Functions

- HeightMap (std::string fileName)

  Constructs a new HeightMap object.
- ∼HeightMap ()

  Destroys the HeightMap object.
- int getFaces () const

  Returns the number of faces of object This function returns number of faces of object specified by HeighMap.

3.3.1 Detailed Description

A class that represents a height map.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 HeightMap()

```
HeightMap::HeightMap (
            std::string fileName )
```

Constructs a new HeightMap object.

Parameters

fileName	The name of the file to read the height map data from.

3.3.2.2 ∼HeightMap()

```
HeightMap::∼HeightMap ( )
```

Destroys the HeightMap object.

3.3.3 Member Function Documentation

3.3.3.1 getFaces()

```
int HeightMap::getFaces ( ) const
```

Returns the number of faces of object This function returns number of faces of object specified by HeighMap.

Returns

The number of faces of object

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

- HeightMap.hpp
- HeightMap.cpp

Chapter 4

File Documentation

4.1 Cube.cpp File Reference

```
#include "Cube.hpp"
#include <iostream>
#include <queue>
```

4.2 Cube.hpp File Reference

```
#include "types.hpp"
#include <string>
```

Classes

- class Cube

  A class that represents a cube.

4.3 Cube.hpp

Go to the documentation of this file.

```
#pragma once
#include "types.hpp"
#include <string>

class Cube
{
private:
    std::vector<Cube *> neighbors;
    Status status = Status::Unchecked;

    int x;
    int y;
    int z;

public:
    Cube(int x, int y, int z);

    void addNeighbor(Cube *cube, Face face);
    int walkThroughNeighbors(Axis axis);
    std::vector<Cube *> getNeighborsAtAxis(Axis axis);
    void setStatus(Status status);

    int getX() const;
    int getY() const;
    int getZ() const;
    Status getStatus() const;
    std::string toString() const;
};
```

4.4 HeightMap.cpp File Reference

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

4.5 HeightMap.hpp File Reference

```
#include <string>
#include <fstream>
#include "Cube.hpp"
```

Classes

- class HeightMap

  A class that represents a height map.

4.6 HeightMap.hpp

Go to the documentation of this file.

```
#pragma once
#include <string>
#include <fstream>
#include "Cube.hpp"

class HeightMap
{
private:
    std::vector<Cube *> cubes;
    int width;
    int length;
    int height;

    void setCube(int x, int y, int z, Cube *cube);
    int getMaxHeight(std::ifstream &file);
    void createCubes(std::ifstream &file);
    void addNeighbors();

    Cube *getCube(int x, int y, int z) const;
    int getFacesAtAxis(Axis axis) const;
    void resetStatus() const;
    int getIndex(int x, int y, int z) const;

public:
    HeightMap(std::string fileName);
    ~HeightMap();

    int getFaces() const;
};
```

4.7 main.cpp File Reference

```
#include <iostream>
#include <string>
#include <chrono>
#include "HeightMap.hpp"
```

Functions

- int main ()

4.7.1 Function Documentation

4.7.1.1 main()

```
int main ( )
```

Author

Patrik Mintěl

Date

21.11.2023

Version

0.3.0 https://patrick115.eu

4.8 types.cpp File Reference

```
#include "types.hpp"
```

Functions

- std::vector< Face > getFacesAtAxis (Axis axis)
- CubeScanData getCubeDataAtAxis (Axis axis)
- std::vector< Axis > getAllAxies ()

4.8.1 Function Documentation

4.8.1.1 getAllAxies()

```
std::vector< Axis > getAllAxies ( )
```

Retrieves all the axies.

Returns

A vector containing all the axies.

4.8.1.2 getCubeDataAtAxis()

```
CubeScanData getCubeDataAtAxis (
             Axis axis )
```

Retrieves the CubeScanData at the specified axis.

Parameters

axis	The axis to retrieve the CubeScanData from.

Returns

The CubeScanData at the specified axis.

4.8.1.3 getFacesAtAxis()

```
std::vector< Face > getFacesAtAxis (
            Axis axis )
```

Retrieves the faces at the specified axis.

Parameters

axis	The axis to retrieve the faces from.

Returns

A vector of Face objects representing the faces at the specified axis.

4.9 types.hpp File Reference

```
#include <vector>
```

Classes

- struct CubeScanData

  A struct that represents the data of the scanning of a cube from specific axis.

Enumerations

- enum Face {
  FRONT , BACK , LEFT , RIGHT ,
  TOP , BOTTOM }

  An enum that represents the faces of a cube.
- enum Axis {
  X , Y , Z , XInvert ,
  YInvert , ZInvert }

  An enum that represents the axies which the cubes are scanned at.
- enum Status { Unchecked , Checked }

  An enum that represents the status of a cube.

Functions

- std::vector< Face > getFacesAtAxis (Axis axis)
- CubeScanData getCubeDataAtAxis (Axis axis)
- std::vector< Axis > getAllAxies ()

4.9.1 Enumeration Type Documentation

4.9.1.1 Axis

```
enum Axis
```

An enum that represents the axies which the cubes are scanned at.

Enumerator

X	
Y	
Z	
XInvert	
YInvert	
ZInvert	

4.9.1.2 Face

```
enum Face
```

An enum that represents the faces of a cube.

Enumerator

FRONT	
BACK	
LEFT	
RIGHT	
TOP	
BOTTOM	

4.9.1.3 Status

```
enum Status
```

An enum that represents the status of a cube.

Enumerator

Unchecked	
Checked	

4.9.2 Function Documentation

4.9.2.1 getAllAxies()

```
std::vector< Axis > getAllAxies ( )
```

Retrieves all the axies.

Returns

A vector containing all the axies.

4.9.2.2 getCubeDataAtAxis()

```
CubeScanData getCubeDataAtAxis (
             Axis axis )
```

Retrieves the CubeScanData at the specified axis.

Parameters

axis	The axis to retrieve the CubeScanData from.

Returns

The CubeScanData at the specified axis.

4.9.2.3 getFacesAtAxis()

```
std::vector< Face > getFacesAtAxis (
             Axis axis )
```

Retrieves the faces at the specified axis.

Parameters

axis	The axis to retrieve the faces from.

Returns

A vector of Face objects representing the faces at the specified axis.

4.10 types.hpp

Go to the documentation of this file.

```
#pragma once
#include <vector>
```

```

enum Face
{
    FRONT,
    BACK,
    LEFT,
    RIGHT,
    TOP,
    BOTTOM
};

enum Axis
{
    X,
    Y,
    Z,
    XInvert,
    YInvert,
    ZInvert
};

enum Status
{
    Unchecked,
    Checked
};

struct
{
    std::vector<Face> neighborsAtFaces;
    Face checkIfFaceHasNeighbor;
} typedef CubeScanData;

std::vector<Face> getFacesAtAxis(Axis axis);
CubeScanData getCubeDataAtAxis(Axis axis);
std::vector<Axis> getAllAxies();
```

Index

~HeightMap
    HeightMap, 10

addNeighbor
    Cube, 6
Axis
    types.hpp, 16

BACK
    types.hpp, 16
BOTTOM
    types.hpp, 16

Checked
    types.hpp, 16
checkIfFaceHasNeighbor
    CubeScanData, 9
Cube, 5
    addNeighbor, 6
    Cube, 6
    getNeighborsAtAxis, 6
    getStatus, 7
    getX, 7
    getY, 7
    getZ, 7
    setStatus, 7
    toString, 8
    walkThroughNeighbors, 8
Cube.cpp, 11
Cube.hpp, 11
CubeScanData, 8
    checkIfFaceHasNeighbor, 9
    neighborsAtFaces, 9

Face
    types.hpp, 16
FRONT
    types.hpp, 16

getAllAxies
    types.cpp, 13
    types.hpp, 17
getCubeDataAtAxis
    types.cpp, 13
    types.hpp, 17
getFaces
    HeightMap, 10
getFacesAtAxis
    types.cpp, 15
    types.hpp, 17
getNeighborsAtAxis
    Cube, 6
getStatus
    Cube, 7
getX
    Cube, 7
getY
    Cube, 7
getZ
    Cube, 7

HeightMap, 9
    ~HeightMap, 10
    getFaces, 10
    HeightMap, 9
HeightMap.cpp, 12
HeightMap.hpp, 12

LEFT
    types.hpp, 16

main
    main.cpp, 13
main.cpp, 13
    main, 13

neighborsAtFaces
    CubeScanData, 9

RIGHT
    types.hpp, 16

setStatus
    Cube, 7
Status
    types.hpp, 16

TOP
    types.hpp, 16
toString
    Cube, 8
types.cpp, 13
    getAllAxies, 13
    getCubeDataAtAxis, 13
    getFacesAtAxis, 15
types.hpp, 15
    Axis, 16
    BACK, 16
    BOTTOM, 16
    Checked, 16
    Face, 16
    FRONT, 16

getAllAxies, 17
getCubeDataAtAxis, 17
getFacesAtAxis, 17
LEFT, 16
RIGHT, 16
Status, 16
TOP, 16
Unchecked, 16
X, 16
XInvert, 16
Y, 16
YInvert, 16
Z, 16
ZInvert, 16

Unchecked
types.hpp, 16

walkThroughNeighbors
Cube, 8

X
types.hpp, 16
XInvert
types.hpp, 16

Y
types.hpp, 16
YInvert
types.hpp, 16

Z
types.hpp, 16
ZInvert
types.hpp, 16