ALG01-semProject

1.0

Generated by Doxygen 1.10.0

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 Map Class Reference

Map class used for load, store and return path find results.

```
#include <Map.h>
```

Public Member Functions

- Map (int rows, int cols)

    Map object constructor for specified size.
- ∼Map ()

    Destructor for deleting all objects in map object.
- Point ∗ GetStart ()

    Start point getter.
- Point ∗ GetFinish ()

    End point getter.
- void AddPoint (Point ∗point)

    Loads point into map.
- Point ∗ GetPoint (int x, int y)

    Getting point from map.
- std::string PrintMap ()

    Print actual state of Map.
- void CalculatePath ()

    Calculating shortest path Implementation Breadth-first search for each neighbour Ends when reach end or check all points.
- std::vector< Point ∗ > GetPath ()

    Return results from calculated path.

Static Public Member Functions

- static Map ∗ LoadFromFile (const std::string &filename)

    Reading map version from file.

3.1.1 Detailed Description

Map class used for load, store and return path find results.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Map()

```
Map::Map (
             int rows,
             int cols )
```

Map object constructor for specified size.

Parameters

rows	Row count
cols	Column count

3.1.2.2 ∼Map()

```
Map::∼Map ( )
```

Destructor for deleting all objects in map object.

3.1.3 Member Function Documentation

3.1.3.1 AddPoint()

```
void Map::AddPoint (
             Point * point )
```

Loads point into map.

Parameters

point	point to be added

3.1.3.2 CalculatePath()

```
void Map::CalculatePath ( )
```

Calculating shortest path Implementation Breadth-first search for each neighbour Ends when reach end or check all points.

3.1.3.3 GetFinish()

```
Point * Map::GetFinish ( )
```

End point getter.

Returns

finish point

3.1.3.4 GetPath()

```
std::vector< Point * > Map::GetPath ( )
```

Return results from calculated path.

Returns

array of Points from start to end

3.1.3.5 GetPoint()

```
Point * Map::GetPoint (
            int x,
            int y )
```

Getting point from map.

Parameters

x	column index in map
y	row index in map

Returns

selected point from map

3.1.3.6 GetStart()

```
Point * Map::GetStart ( )
```

Start point getter.

Returns

start point

3.1.3.7 LoadFromFile()

```
Map * Map::LoadFromFile (
             const std::string & filename )  [static]
```

Reading map version from file.

Parameters

filename	path to file with map input

Returns

new Map generated from text

3.1.3.8 PrintMap()

```
std::string Map::PrintMap ( )
```

Print actual state of Map.

Returns

console printable string

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

- Map.h
- Map.cpp

3.2 Point Class Reference

Point class which is composite for Map Point represents single place in map.

```
#include <Point.h>
```

Public Member Functions

- Point (int x, int y, char level)

  Creates nnew point.
- ∼Point ()

  Destroys actual point and its refferences.
- int GetX ()

  Getter for x directory.
- int GetY ()

  Getter for y directory.
- int GetDistance ()

Getter for distance.
- char GetLevel ()

  Getter for level replacing start with 'a' value and end with 'z'.
- char PrintLevel ()

  Getter for level with real char value from map.
- void SetDistance (int distance)

  Setter for distance.
- void AddPrevious (Point ∗point)

  Method for adding refference to previous Point.
- Point ∗ GetPrevious ()

  Getter for actual stored previous Point.

3.2.1 Detailed Description

Point class which is composite for Map Point represents single place in map.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Point()

```
Point::Point (
             int x,
             int y,
             char level )
```

Creates nnew point.

Parameters

x	

3.2.2.2 ∼Point()

```
Point::∼Point ( )
```

Destroys actual point and its refferences.

3.2.3 Member Function Documentation

3.2.3.1 AddPrevious()

```
void Point::AddPrevious (
             Point ∗ point )
```

Method for adding refference to previous Point.

Parameters

point	

3.2.3.2 GetDistance()

```
int Point::GetDistance ( )
```

Getter for distance.

Returns

distance as integer

3.2.3.3 GetLevel()

```
char Point::GetLevel ( )
```

Getter for level replacing start with 'a' value and end with 'z'.

Returns

3.2.3.4 GetPrevious()

```
Point * Point::GetPrevious ( )
```

Getter for actual stored previous Point.

Returns

pointer to previous Point

3.2.3.5 GetX()

```
int Point::GetX ( )
```

Getter for x directory.

Returns

index x

3.2.3.6 GetY()

```
int Point::GetY ( )
```

Getter for y directory.

Returns

index y

3.2.3.7 PrintLevel()

```
char Point::PrintLevel ( )
```

Getter for level with real char value from map.

Returns

3.2.3.8 SetDistance()

```
void Point::SetDistance (
             int distance )
```

Setter for distance.

Parameters

distance	

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

- Point.h
- Point.cpp

Chapter 4

File Documentation

4.1 ALG01-semProject.cpp File Reference

```
#include <iostream>
#include <fstream>
#include <string>
#include "Map.h"
```

Functions

- int main ()

  Solution for shortest-path find.

4.1.1 Function Documentation

4.1.1.1 main()

```
int main ( )
```

Solution for shortest-path find.

Returns

4.2 Map.cpp File Reference

```
#include "Map.h"
#include <iostream>
#include <queue>
```

4.3 Map.h File Reference

```
#include "Point.h"
#include <string>
#include <vector>
#include <iostream>
#include <fstream>
```

Classes

- class Map

  Map class used for load, store and return path find results.

4.4 Map.h

Go to the documentation of this file.

```
#pragma once
#include "Point.h"
#include <string>
#include <vector>
#include <iostream>
#include <fstream>
class Map
{
private:
    std::vector<Point*> mapArray;
    Point* start = nullptr;
    Point* finish = nullptr;
    int rows;
    int cols;
    Point* ProcessPoint(Point* actual, Point* next);
public:
    Map(int rows, int cols);
    ~Map();
    Point* GetStart();
    Point* GetFinish();
    void AddPoint(Point* point);
    Point* GetPoint(int x, int y);
    std::string PrintMap();
    void CalculatePath();
    std::vector<Point*> GetPath();
    static Map* LoadFromFile(const std::string& filename);


};
```

4.5 Point.cpp File Reference

```
#include "Point.h"
```

4.6 Point.h File Reference

Classes

- class Point

  Point class which is composite for Map Point represents single place in map.

4.7 Point.h

Go to the documentation of this file.

```
#pragma once
class Point
{
private:
    int x;
    int y;
    char level;
    int distance = 2147483647;//int maxvalue
    Point* previous = nullptr;
public:
    Point(int x, int y, char level);
    ~Point();
    int GetX();
    int GetY();
    int GetDistance();
    char GetLevel();
    char PrintLevel();
    void SetDistance(int distance);
    void AddPrevious(Point* point);
    Point* GetPrevious();


};
```

Index