My Project

Generated by Doxygen 1.8.16

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 Maze Class Reference

Public Member Functions

- Maze ()
- Maze (std::size_t rows, std::size_t cols, std::size_t begin, std::size_t end) throw (std::invalid_argument)
- bool set (const std::vector< unsigned char > &maze, std::size_t rows)
- const std::vector< unsigned char > & get () const
- bool create (std::size_t rows, std::size_t cols, std::size_t begin, std::size_t end)
- bool solve ()
- bool read (std::string filename)
- bool write (std::string filename) const

Friends

- std::ostream & operator<< (std::ostream &os, const Maze &m)

2.1.1 Constructor & Destructor Documentation

2.1.1.1 Maze() [1/2]

`Maze::Maze ( )`

Create a blank maze of size 0-by-0

Returns

maze with rows=0 and cols=0

2.1.1.2 Maze() [2/2]

```
Maze::Maze (
            std::size_t rows,
            std::size_t cols,
            std::size_t begin,
            std::size_t end ) throw ( std::invalid_argument)
```

Constructor will generate a random maze given input parameters. Throws exception if maze parameters imply/yield an invalid maze.

Parameters

rows	- height of maze
cols	- width of maze
begin	- entry point in maze
end	- exit point in maze

2.1.2 Member Function Documentation

2.1.2.1 create()

```
bool Maze::create (
            std::size_t rows,
            std::size_t cols,
            std::size_t begin,
            std::size_t end )
```

Generate a new maze with the given parameters. Should overwrite existing maze, if any.

Parameters

rows	- height of maze
cols	- width of maze
begin	- entry point in maze
end	- exit point in maze

Returns

true if we could create a valid maze

2.1.2.2 get()

```
const std::vector<unsigned char>& Maze::get ( ) const
```

Return the maze

Returns

an unsolved or solved maze, depending on whether it has been solved or not

2.1.2.3 read()

```
bool Maze::read (
            std::string filename )
```

Read a solved or unsolved maze from the PNG file. Should overwrite existing maze, if any

Parameters

filename	- name of PNG file containing unsolved or solved maze

Returns

whether the maze could be read and is valid

2.1.2.4 set()

```
bool Maze::set (
            const std::vector< unsigned char > & maze,
            std::size_t rows )
```

Function to set the maze we want to solve, validate it, and set private variables appropriately (a valid maze will only have characters listed in the project description for an unsolved maze). Should overwrite existing maze, if any.

Parameters

maze	- a one dimensional vector representing an unsolved maze
rows	- the number of rows in the maze

Returns

whether the maze is valid

2.1.2.5 solve()

```
bool Maze::solve ( )
```

Find the path from B to E in the current maze and record the path in the maze

Returns

false if no path could be found from B to E or error, true otherwise

2.1.2.6 write()

```
bool Maze::write (
            std::string filename ) const
```

Write the unsolved or solved mazed to a PNG file

Parameters

filename	- name of PNG file to write the maze to

Returns

false if the file can't be written

2.1.3 Friends And Related Function Documentation

2.1.3.1 operator<<

```
std::ostream& operator<< (
            std::ostream & os,
            const Maze & m )  [friend]
```

Operator to print maze to std::cout (useful for debugging)

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

- Maze.hpp

Index