

AutoCell

Generated by Doxygen 1.8.13

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	Cell Class Reference	5
3.1.1	Constructor & Destructor Documentation	5
3.1.1.1	Cell()	5
3.1.2	Member Function Documentation	6
3.1.2.1	addNeighbour()	6
3.1.2.2	getNeighbours()	6
3.1.2.3	getState()	6
3.1.2.4	setState()	6
3.1.2.5	validState()	7
4	File Documentation	9
4.1	cell.cpp File Reference	9
4.2	cell.h File Reference	9
4.3	main.cpp File Reference	9
4.3.1	Function Documentation	9
4.3.1.1	main()	9
	Index	11

Chapter 1

Class Index

1.1 Class List

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

Cell	5
--------------------------------	-------------------

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

cell.cpp	9
cell.h	9
main.cpp	9

Chapter 3

Class Documentation

3.1 Cell Class Reference

```
#include <cell.h>
```

Public Member Functions

- [Cell](#) (unsigned int state=0)
Constructs a cell with the state given. State 0 is dead state.
- void [setState](#) (unsigned int state)
Set temporary state.
- void [validState](#) ()
Validate temporary state.
- unsigned int [getState](#) () const
Access current cell state.
- bool [addNeighbour](#) (const [Cell](#) *neighbour)
Add a new neighbour to the [Cell](#).
- QVector< const [Cell](#) * > [getNeighbours](#) () const
Access neighbours list.

3.1.1 Constructor & Destructor Documentation

3.1.1.1 Cell()

```
Cell::Cell (
    unsigned int state = 0 )
```

Constructs a cell with the state given. State 0 is dead state.

Parameters

<i>state</i>	Cell state, dead state by default
--------------	---

3.1.2 Member Function Documentation

3.1.2.1 addNeighbour()

```
bool Cell::addNeighbour (
    const Cell * neighbour )
```

Add a new neighbour to the [Cell](#).

Parameters

<i>neighbour</i>	New neighbour
------------------	---------------

Returns

False if the neighbour already exists

3.1.2.2 getNeighbours()

```
QVector< const Cell * > Cell::getNeighbours ( ) const
```

Access neighbours list.

3.1.2.3 getState()

```
unsigned int Cell::getState ( ) const
```

Access current cell state.

3.1.2.4 setState()

```
void Cell::setState (
    unsigned int state )
```

Set temporary state.

To change current cell state, use [setState\(unsigned int state\)](#) then [validState\(\)](#).

Parameters

<i>state</i>	New state
--------------	-----------

3.1.2.5 validState()

```
void Cell::validState ( )
```

Validate temporary state.

To change current cell state, use [setState\(unsigned int state\)](#) then [validState\(\)](#).

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

- [cell.h](#)
- [cell.cpp](#)

Chapter 4

File Documentation

4.1 cell.cpp File Reference

```
#include "cell.h"
```

4.2 cell.h File Reference

```
#include <QVector>
```

Classes

- class [Cell](#)

4.3 main.cpp File Reference

```
#include <QApplication>
#include <QDebug>
#include "cell.h"
```

Functions

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

4.3.1 Function Documentation

4.3.1.1 main()

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


Index

addNeighbour
Cell, [6](#)

Cell, [5](#)
addNeighbour, [6](#)
Cell, [5](#)
getNeighbours, [6](#)
getState, [6](#)
setState, [6](#)
validState, [7](#)

cell.cpp, [9](#)

cell.h, [9](#)

getNeighbours
Cell, [6](#)

getState
Cell, [6](#)

main
main.cpp, [9](#)
main.cpp, [9](#)
main, [9](#)

setState
Cell, [6](#)

validState
Cell, [7](#)