

Assignment 4

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	Board Class Reference	5
3.1.1	Member Function Documentation	5
3.1.1.1	getCell()	5
3.2	Cell Class Reference	6
3.2.1	Detailed Description	6
3.2.2	Constructor & Destructor Documentation	6
3.2.2.1	Cell()	6
3.2.3	Member Function Documentation	7
3.2.3.1	setAdj()	7
3.2.3.2	setState()	7
3.3	View Class Reference	7
3.3.1	Detailed Description	8
3.3.2	Member Function Documentation	8
3.3.2.1	view()	8
3.3.2.2	writeState()	8

4	File Documentation	9
4.1	include/Cell.h File Reference	9
4.1.1	Detailed Description	9
4.2	include/CellGrid.h File Reference	9
4.2.1	Detailed Description	10
4.3	include/CellTypes.h File Reference	10
4.3.1	Detailed Description	10
4.4	include/GameBoard.h File Reference	10
4.4.1	Detailed Description	11
4.5	include/View.h File Reference	11
4.5.1	Detailed Description	11
	Index	13

Chapter 1

Class Index

1.1 Class List

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

Board	5
Cell		
	Cell abstract data type	6
View		
	View module for viewing and outputting the state of the game	7

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

include/ Cell.h	9
include/ CellGrid.h	9
include/ CellTypes.h	10
include/ GameBoard.h GameBoard module for Conway's Game of Life - tracks state of gameboard	10
include/ View.h	11

Chapter 3

Class Documentation

3.1 Board Class Reference

Public Member Functions

- [Board](#) ()
Empty constructor for a board.
- [Board](#) (const char *inFile)
Constructor for a board.
- [Cell](#) [getCell](#) (nat x, nat y) const
Getter - gets cell at x and y coordinate.
- void [nextState](#) ()
Getter - gets the next state of the game.
- void [outputAndView](#) ()
Calls the [View\(\)](#) function to output and view game state.

3.1.1 Member Function Documentation

3.1.1.1 [getCell\(\)](#)

```
Cell Board::getCell (  
    nat x,  
    nat y ) const
```

Getter - gets cell at x and y coordinate.

Parameters

<i>x</i>	- the x coordinate of the cell
<i>y</i>	- the y coordinate of the cell

Returns

returns a cell

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

- include/[GameBoard.h](#)

3.2 Cell Class Reference

[Cell](#) abstract data type.

```
#include <Cell.h>
```

Public Member Functions

- [Cell](#) ()
Empty constructor for a cell.
- [Cell](#) ([nat](#) n0, [nat](#) n1)
Empty constructor for a cell.
- [nat](#) [getX](#) () const
Gets the x coordinate for a cell.
- [nat](#) [getY](#) () const
Gets the y coordinate for a cell.
- void [setState](#) ([CellState](#) s)
Sets the state for a cell.
- [CellState](#) [getState](#) () const
Gets the state for a cell.
- [nat](#) [getAdj](#) () const
Gets the number of adjacent cells for a cell.
- void [setAdj](#) ([nat](#) n)
Sets the number of adjacent cells for a cell.

3.2.1 Detailed Description

[Cell](#) abstract data type.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 [Cell](#)()

```
Cell::Cell (  
    nat n0,  
    nat n1 )
```

Empty constructor for a cell.

Parameters

<i>n0</i>	- the x coordinate of the cell
<i>n1</i>	- the y coordinate of the cell

3.2.3 Member Function Documentation

3.2.3.1 setAdj()

```
void Cell::setAdj (
    nat n )
```

Sets the number of adjacent cells for a cell.

Parameters

<i>n</i>	- the number of neighbours/adjacents for the cell
----------	---

3.2.3.2 setState()

```
void Cell::setState (
    CellState s )
```

Sets the state for a cell.

Parameters

<i>s</i>	- the state to be set for the cell
----------	------------------------------------

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

- include/[Cell.h](#)

3.3 View Class Reference

[View](#) module for viewing and outputting the state of the game.

```
#include <View.h>
```

Public Member Functions

- void [view](#) ([CellGrid](#) grid)
[View](#) module for viewing game in state.
- void [writeState](#) ([CellGrid](#) grid)
write module for writing the state of the game to an output file

3.3.1 Detailed Description

[View](#) module for viewing and outputting the state of the game.

3.3.2 Member Function Documentation

3.3.2.1 view()

```
void View::view (  
    CellGrid grid )
```

[View](#) module for viewing game in state.

Parameters

<i>grid</i>	- a grid of cells
-------------	-------------------

3.3.2.2 writeState()

```
void View::writeState (  
    CellGrid grid )
```

write module for writing the state of the game to an output file

Parameters

<i>grid</i>	- a grid of cells
-------------	-------------------

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

- include/[View.h](#)

Chapter 4

File Documentation

4.1 include/Cell.h File Reference

```
#include "CellTypes.h"
```

Classes

- class [Cell](#)
Cell abstract data type.

4.1.1 Detailed Description

Author

Leon So | macid: sol4

Date

2019-04-13

4.2 include/CellGrid.h File Reference

```
#include "CellTypes.h"  
#include "Cell.h"  
#include <vector>
```

Typedefs

- typedef std::vector< std::vector< [Cell](#) > > [CellGrid](#)
Describes a 2D grid of cells.

4.2.1 Detailed Description

Author

Leon So | macid: sol4

Date

2019-04-13

4.3 include/CellTypes.h File Reference

Typedefs

- typedef unsigned int [nat](#)
Describes a natural number.

Enumerations

- enum [CellState](#) { **Dead**, **Alive** }
Describes the suit of a card.

4.3.1 Detailed Description

Author

Leon So | macid: sol4

Date

2019-04-13

4.4 include/GameBoard.h File Reference

GameBoard module for Conway's Game of Life - tracks state of gameboard.

```
#include "CellTypes.h"
#include "Cell.h"
#include "CellGrid.h"
```

Classes

- class [Board](#)

4.4.1 Detailed Description

GameBoard module for Conway's Game of Life - tracks state of gameboard.

Author

Leon So | macid: sol4

Date

2019-04-13

4.5 include/View.h File Reference

```
#include "CellTypes.h"
#include "GameBoard.h"
#include "Cell.h"
#include "CellGrid.h"
```

Classes

- class [View](#)
[View](#) module for viewing and outputting the state of the game.

4.5.1 Detailed Description

Author

Leon So | macid: sol4

Date

2019-04-13

Index

- Board, [5](#)
 - [getCell](#), [5](#)
- Cell, [6](#)
 - [Cell](#), [6](#)
 - [setAdj](#), [7](#)
 - [setState](#), [7](#)
- [getCell](#)
 - [Board](#), [5](#)
- [include/Cell.h](#), [9](#)
- [include/CellGrid.h](#), [9](#)
- [include/CellTypes.h](#), [10](#)
- [include/GameBoard.h](#), [10](#)
- [include/View.h](#), [11](#)
- [setAdj](#)
 - [Cell](#), [7](#)
- [setState](#)
 - [Cell](#), [7](#)
- [View](#), [7](#)
 - [view](#), [8](#)
 - [writeState](#), [8](#)
- [view](#)
 - [View](#), [8](#)
- [writeState](#)
 - [View](#), [8](#)