Game of Life

0.01

Generated by Doxygen 1.7.4

Tue May 31 2011 14:34:53

Contents

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 documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 Cell Class Reference

`#include <GameOfLife.h>`

Public Member Functions

- void kill ()
- void revive ()
- bool isAlive ()

3.1.1 Detailed Description

Cell class definition

The cell class represents a Cell in the game of life. It cell has a state and methods for killing or reviving.

3.1.2 Member Function Documentation

3.1.2.1 bool Cell::isAlive ( )

Verifies whether a cell is alive or not

3.1.2.2 void Cell::kill ( )

Changes the state of a cell to DEAD

3.1.2.3 void Cell::revive ( )

Changes the state of acell to ALIVE

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

- src/include/GameOfLife.h
- src/cpp/GameOfLife.cpp

3.2 Controller Class Reference

`#include <Controller.h>`

Public Member Functions

- Controller (GameOfLife &g, GameBoard &b)
- void StartGame ()

3.2.1 Detailed Description

Controller class definition

This class deals with the user inputs, and based on the user actions it triggers the proper functionality.

Besides that, its interface is really simple. Only one public method is available, which just start a game.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Controller::Controller (GameOfLife & *g*, GameBoard & *b*) [inline]

Controller constructor, taking a game instance (model) and a game board (view)

3.2.3 Member Function Documentation

3.2.3.1 void Controller::StartGame ( )

Start the execution of the game

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

- src/include/Controller.h
- src/cpp/Controller.cpp

3.3 GameBoard Class Reference

`#include <GameBoard.h>`

Public Member Functions

- GameBoard (int r, int c)
- void update (GameOfLife &game)

3.3.1 Detailed Description

GameBoard class definition

This class represents a basic game of life environment, which comprises a bidimensional grid. New environments shuld be implemented.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 GameBoard::GameBoard (int *r*, int *c*) [inline]

basic constructor, taking the number of rows and columns

3.3.3 Member Function Documentation

3.3.3.1 void GameBoard::update (GameOfLife & *game*)

updates the environment representation

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

- src/include/GameBoard.h
- src/cpp/GameBoard.cpp

3.4 GameOfLife Class Reference

`#include <GameOfLife.h>`

Public Member Functions

- GameOfLife (int w, int h)
- int aliveCells ()
- int aliveNeighborCells (int w, int h)
- bool isCellAlive (int w, int h)

- void makeCellAlive (int w, int h)
- void makeCellDead (int w, int h)
- void nextGeneration ()
- int getWidth () const
- int getHeight () const

3.4.1 Detailed Description

GameOfLife class definition

Modularizes the behavior of a game of life. In this version, just one algorithm for evolving the environment for a next generation is allowd. To solve this provlem, we could evolve this design using either the Strategy or Template Method design patterns (both).

This version is not well designed, since it keeps reference to one instance of the Statistics class. It would be nice to see an implementation of this class being a subject participant of the Observer pattern, whereas the Statics class could be one subscriber participant of the Observer pattern.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 GameOfLife::GameOfLife (int *w,* int *h*)

Constructor, taking the number of columns and rows

3.4.3 Member Function Documentation

3.4.3.1 int GameOfLife::aliveCells ( )

Returns the number of cells in the ALIVE state

3.4.3.2 int GameOfLife::aliveNeighborCells (int *w,* int *h*)

Given the position of a cell, returns the number of alive neighbors

3.4.3.3 int GameOfLife::getHeight ( ) const [inline]

Access method to the number of rows

3.4.3.4 int GameOfLife::getWidth ( ) const [inline]

Access method to the number of columns

3.4.3.5 bool GameOfLife::isCellAlive (int *w*, int *h*)

Checks whether a cell is alive

3.4.3.6 void GameOfLife::makeCellAlive (int *w*, int *h*)

Makes a given cell alive

3.4.3.7 void GameOfLife::makeCellDead (int *w*, int *h*)

Kills a given cell

3.4.3.8 void GameOfLife::nextGeneration ( )

Leads the game state to a next generation

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

- src/include/GameOfLife.h
- src/cpp/GameOfLife.cpp

3.5 GameOfLifeTest Class Reference

Public Member Functions

- void **setUp** ()
- void **tearDown** ()
- void **testCellConstructor** ()
- void **testKill** ()
- void **testRevive** ()
- void **testGameOfLifeConstructor** ()
- void **testAliveCells** ()
- void **testMakeCellAlive** ()
- void **testMakeCellDead** ()
- void **testAliveNeighborCells** ()

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

- src/include/GameOfLifeTest.h
- src/tests/GameOfLifeTest.cpp

3.6 Statistics Class Reference

```
#include <Statistics.h>
```

Public Member Functions

- Statistics ()
- void survive ()
- void kill ()

3.6.1 Detailed Description

Statistics class definition

This class keeps a history of the number of killed and revived cells.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 Statistics::Statistics ( ) [inline]

Basic constructor, initializing the number of survivors and killed cells to zero

3.6.3 Member Function Documentation

3.6.3.1 void Statistics::kill ( )

notified that a cell was killed

3.6.3.2 void Statistics::survive ( )

Notifies that a cell was revived

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

- src/include/Statistics.h
- src/cpp/Statistics.cpp

Chapter 4

File Documentation

4.1 src/include/Controller.h File Reference

```
#include "GameOfLife.h"
#include "GameBoard.h"
```

Classes

- class Controller

4.1.1 Detailed Description

A controller component of the GameOfLife game Rodrigo Bonifacio (rbonifacio[at]cic.unb.br)

4.2 src/include/GameBoard.h File Reference

```
#include <string>
#include "GameOfLife.h"
```

Classes

- class GameBoard

Variables

- const string **line** = "+-----+"

- const string **dead** = "| |"
- const string **alive** = "| o |"

4.2.1 Detailed Description

A game board to display a GameOfLife environment. This is one of the UI modules.

Rodrigo Bonifacio (rbonifacio[at]cic.unb.br)

4.3 src/include/GameOfLife.h File Reference

```
#include "Statistics.h"
```

Classes

- class Cell
- class GameOfLife

Enumerations

- enum EnumState { **DEAD**, **ALIVE** }

  EnumState enumeration.

4.3.1 Detailed Description

The game of life model component, which compromises the Cell and GameOfLiife class definitions.

Rodrigo Bonifacio (rbonifacio[at]cic.unb.br)

4.3.2 Enumeration Type Documentation

4.3.2.1 enum EnumState

EnumState enumeration.

Define the valid states of a cell.

4.4 src/include/Statistics.h File Reference

Classes

- class Statistics

4.4.1 Detailed Description

A component that keeps information about the number of born and killed cells. It is another UI interface, since each time a cell is killed or revived, a text is displayed.

Rodrigo Bonifacio (rbonifacio[at]cic.unb.br)

Index