The Game Of Life

AUTHOR: Wojciech Halber
Version 1.0
Tue May 25 2021

Main Page

Main funcion, just using the **menu.h** class

Author

Wojciech Halber

Date

25.05.2021

Version

1.0

Kontakt:

wojciech.halber@pw.edu.pl

Class Index

Class List

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

File Index

File List

Here is a list of all files with brief descriptions:

Class Documentation

board Class Reference

The board class is made of cells which may be dead or alive.
`#include <board.h>`

Public Member Functions

- void **makeBoard** ()
  makeBoard

- void **deleteBoard** ()
  deleteBoard

- void **printBoard** ()
  printBoard

- void **setAlive** (int x, int y)
  Funtion switch cells statement from dead to alive or opposite way.

- bool **getAlive** (int x, int y)
  Sets if chosen cell is alive (1) or dead (0)

- int **getHeight** () const
  Returns if chosen cell is alive (1) or dead (0)

- void **setHeight** ()
  Returns board's height.

- int **getWidth** () const
  Sets board's height.

- void **setWidth** ()
  Returns board's width.

- **cell** * **getUpLeft** () const
  Sets board's width.

- void **addColumn** (bool place)
  Returns address of first cell.

- void **deleteColumn** (bool place)
  Adds first (0) or last (1) column.

- void **addLine** (bool place)
  Deletes first (0) or last (1) column.

- void **deleteLine** (bool place)
  Adds first (0) or last (1) line.

- void **nextGen** ()
  Deletes first (0) or last (1) line.

- std::string **getGameRule** () const
  Funtion returns actuall gamerule.

- void **setGameRule** (const std::string &value)
  Funtions change gamerule.

- int **neibAlive** (int x, int y)
  Funtions reurns number of neighbors alive.

- void **border** ()
  Checks, if there any alive cells on board's borders, then adds new columns or lines to avoid getting alive cells outta map.

Detailed Description

The board class is made of cells which may be dead or alive.

Technically, it's 4-way list made by cell class objects. Ever cell has it's own (x,y) coords counted from 0 and can be dead (0) or alive (1). You can make many different boards at the same time

Warning

The class wouldn't work without **cell.h**

Member Function Documentation

void board::addColumn (bool *place*)

Returns address of first cell.

Funtion adds new column to existing board at the front or end

Parameters

place	Switch for place for new column. 0 adds at the front, 1 at the end

void board::addLine (bool *place*)

Deletes first (0) or last (1) column.

Funtion adds new line to existing board at the front or end

Parameters

place	Switch for place for new line. 0 adds at the front, 1 at the end

void board::border ()

Checks, if there any alive cells on board's borders, then adds new columns or lines to avoid getting alive cells outta map.

void board::deleteBoard ()

deleteBoard

Class destructor

void board::deleteColumn (bool *place*)

Adds first (0) or last (1) column.

Deletes first or last column of board

Parameters

place	Switch for first (0) or last (1) column

void board::deleteLine (bool *place*)

Adds first (0) or last (1) line.

Deletes first or last line of board

Parameters

place	Switch for first (0) or last (1) line

bool board::getAlive (int *x*, int *y*)

Sets if chosen cell is alive (1) or dead (0)

Function return if the cell is alive (1) or dead (0)

Parameters

x	X coord of choosen cell
y	Y coord of choosen cell

Returns

Returns boolean 0 if cell is dead or 1 if alive

std::string board::getGameRule () const

Funtion returns actuall gamerule.

Returns

String XXX/YYY

int board::getHeight () const

Returns if chosen cell is alive (1) or dead (0)

Returns actual height of board

Returns

Integer equal to board height

cell * board::getUpLeft () const

Sets board's width.

Function returns address of first cell (coords 0,0)

Returns

Cell class address

int board::getWidth () const

Sets board's height.

Function returns actual width of board

Returns

Integer equal to board width

void board::makeBoard ()

makeBoard

Class constructor

int board::neibAlive (int *x*, int *y*)

Funtions reurns number of neighbors alive.

Parameters

x	X coord of cell
y	Y coord of cell

Returns

Integer equal to number of neighbors alive

void board::nextGen ()

Deletes first (0) or last (1) line.

Funtion evolve board by one generation

void board::printBoard ()

printBoard

Prints on console existing board

void board::setAlive (int *x*, int *y*)

Funtion switch cells statement from dead to alive or opposite way.

Parameters

x	X coord of choosen cell
y	Y coord of choosen cell

void board::setGameRule (const std::string & *value*)

Funtions change gamerule.

Parameters

value	Must be string cointaining only digits and one '/'

void board::setHeight ()

Returns board's height.

Funtion asks user for wanted board height and sets it

void board::setWidth ()

Returns board's width.

Funtion asks user for wanted board width and sets it

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

- projekt/**board.h**
- projekt/**board.cpp**

cell Class Reference

Cell is object containing bool alive and 4 cell class pointers of its neighbours.
`#include <cell.h>`

Public Member Functions

- void **setAlive** ()
  Switch cell statement from dead to alive or opposite way.

- void **setUp** (**cell** *up)
  Sets pointer Up.

- void **setDown** (**cell** *down)
  Sets pointer Down.

- void **setRight** (**cell** *right)
  Sets pointer Right.

- void **setLeft** (**cell** *left)
  Sets pointer Left.

- bool **ifAlive** ()
  Returns statement of cell.

- **cell** * **getUp** ()
  Returns Up pointer.

- **cell** * **getDown** ()
  Returns Down pointer.

- **cell** * **getRight** ()
  Returns Right pointer.

- **cell** * **getLeft** ()
  Returns Left pointer.

- int **neighboursAlive** ()
  Funcions return nu,ber of alive neighbours.

Detailed Description

Cell is object containing bool alive and 4 cell class pointers of its neighbours.

It's designed to make 4-way lists, for example game board. Cell can be dead (0) or alive (1).

Member Function Documentation

cell * cell::getDown ()

Returns Down pointer.

Returns
Cell pointer

cell * cell::getLeft ()

Returns Left pointer.

Returns
Cell pointer

cell * cell::getRight ()

Returns Right pointer.

Returns
Cell pointer

cell * cell::getUp ()

Returns Up pointer.

Returns
Cell pointer

bool cell::ifAlive ()

Returns statement of cell.

Returns
Boolean, 0 if dead or 1 if alive

int cell::neighboursAlive ()

Funcions return nu,ber of alive neighbours.

Returns
Integer of alive neighbours

void cell::setAlive ()

Switch cell statement from dead to alive or opposite way.

void cell::setDown (cell * *down*)

Sets pointer Down.

Parameters

down	Cell type pointer

void cell::setLeft (cell * *left*)

Sets pointer Left.

Parameters

left	Cell type pointer

void cell::setRight (cell * *right*)

Sets pointer Right.

Parameters

right	Cell type pointer

void cell::setUp (cell * *up*)

Sets pointer Up.

Parameters

up	Cell type pointer

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

- projekt/**cell.h**
- projekt/**cell.cpp**

menu Class Reference

User interface class.
`#include <menu.h>`

Public Member Functions

- **menu** ()
  Main program function.

Detailed Description

User interface class.

Warning

This class needs **board.h** class

Constructor & Destructor Documentation

menu::menu ()

Main program function.

Sets board, print menu options etc.

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

- projekt/**menu.h**
- projekt/**menu.cpp**

File Documentation

projekt/board.cpp File Reference

```
#include "board.h"
#include <iostream>
```

projekt/board.h File Reference

Headline file of class board.

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

Classes

- class **board**
  The board class is made of cells which may be dead or alive.

Detailed Description

Headline file of class board.

Cointains constructor, destructor ans some methods to modify the Board

projekt/cell.cpp File Reference

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

projekt/cell.h File Reference

Headline file of class cell.

Classes

- class **cell**
  Cell is object containing bool alive and 4 cell class pointers of its neighbours.

Detailed Description

Headline file of class cell.

Cointains class constructor, destructor and some methods to modify the cell

projekt/main.cpp File Reference

```
#include <iostream>
#include "board.h"
#include "menu.h"
```

Functions

- int **main** ()

Function Documentation

int main ()

projekt/menu.cpp File Reference

```
#include "menu.h"
```

projekt/menu.h File Reference

Header of UI class.

```
#include "board.h"
#include <ctime>
#include <iostream>
#include <fstream>
```

Classes

- class **menu**
  User interface class.

Detailed Description

Header of UI class.

Index

INDEX