labo_08_gachet_jean_gallay_david

Generated by Doxygen 1.8.13

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

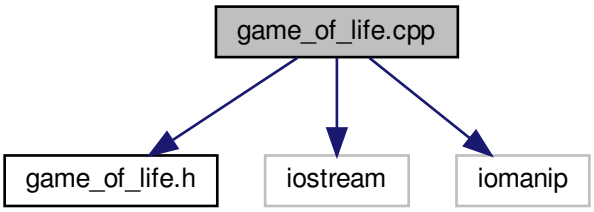
Chapter 2

File Documentation

2.1 game_of_life.cpp File Reference

```
#include "game_of_life.h"
#include <iostream>
#include <iomanip>
```

Include dependency graph for game_of_life.cpp:



Macros

- #define print cout << setw(3)

Functions

- unsigned nbOfNeighbours (bool game[HEIGHT][WIDTH], int line, int column)
- char getDisplayChar (bool value)
- void copyArray (bool fromArray[HEIGHT][WIDTH], bool toArray[HEIGHT][WIDTH])
- void preFillGame (bool game[HEIGHT][WIDTH])
- bool computeNextGen (bool currentGen[HEIGHT][WIDTH])
- void computeMultipleGens (bool currentGen[HEIGHT][WIDTH], unsigned n, bool autoStop)
- void displayGame (const bool game[HEIGHT][WIDTH])

2.1.1 Macro Definition Documentation

2.1.1.1 print

```
#define print cout << setw(3)
```

2.1.2 Function Documentation

2.1.2.1 computeMultipleGens()

```
void computeMultipleGens (
            bool currentGen[HEIGHT][WIDTH],
            unsigned n,
            bool autoStop = true )
```

Take a game and computes the n next generations (stops if a stable state is reached). Displays the game after each generation.

Parameters

currentGen	[IN] current state of the game [OUT] new state of the game
n	number of generations to compute
autoStop	stops if no more changes are detected between generations

2.1.2.2 computeNextGen()

```
bool computeNextGen (
            bool currentGen[HEIGHT][WIDTH] )
```

Take a game and computes the next generation.

Parameters

currentGen	[IN] current state of the game [OUT] new state of the game

Returns

if a change occured

2.1.2.3 copyArray()

```
void copyArray (
            bool fromArray[HEIGHT][WIDTH],
            bool toArray[HEIGHT][WIDTH] )
```

Copys the first array into the second array.

Parameters

fromArray	origin array
toArray	target array

2.1.2.4 displayGame()

```
void displayGame (
            const bool game[HEIGHT][WIDTH] )
```

Display the game.

Parameters

game	

2.1.2.5 getDisplayChar()

```
char getDisplayChar (
            bool value )
```

Returns the char to display depending on the cell's state.

Parameters

value	cell's state

Returns

character to display

2.1.2.6 nbOfNeighbours()

```
unsigned nbOfNeighbours (
            bool game[HEIGHT][WIDTH],
```

```
            int line,
            int column )
```

Compute the number of neighbours of a cell.

Parameters

game	
line	cell's line
column	cell's column

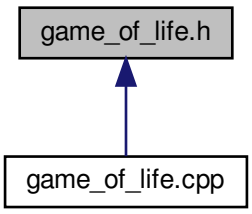
Returns

2.1.2.7 preFillGame()

```
void preFillGame (
            bool game[HEIGHT][WIDTH] )
```

2.2 game_of_life.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define HEIGHT 9
- #define WIDTH 9
- #define ALIVE true
- #define DEAD false
- #define ALIVE_CHAR 'X'
- #define DEAD_CHAR ' '

Functions

- bool computeNextGen (bool currentGen[HEIGHT][WIDTH])
- void computeMultipleGens (bool currentGen[HEIGHT][WIDTH], unsigned n, bool autoStop=true)
- void displayGame (const bool game[HEIGHT][WIDTH])
- void preFillGame (bool game[HEIGHT][WIDTH])

2.2.1 Macro Definition Documentation

2.2.1.1 ALIVE

```
#define ALIVE true
```

2.2.1.2 ALIVE_CHAR

```
#define ALIVE_CHAR 'X'
```

2.2.1.3 DEAD

```
#define DEAD false
```

2.2.1.4 DEAD_CHAR

```
#define DEAD_CHAR ' '
```

2.2.1.5 HEIGHT

```
#define HEIGHT 9
```

2.2.1.6 WIDTH

```
#define WIDTH 9
```

2.2.2 Function Documentation

2.2.2.1 computeMultipleGens()

```
void computeMultipleGens (
            bool currentGen[HEIGHT][WIDTH],
            unsigned n,
            bool autoStop = true )
```

Take a game and computes the n next generations (stops if a stable state is reached). Displays the game after each generation.

Parameters

currentGen	[IN] current state of the game [OUT] new state of the game
n	number of generations to compute
autoStop	stops if no more changes are detected between generations

2.2.2.2 computeNextGen()

```
bool computeNextGen (
            bool currentGen[HEIGHT][WIDTH] )
```

Take a game and computes the next generation.

Parameters

currentGen	[IN] current state of the game [OUT] new state of the game

Returns

if a change occured

2.2.2.3 displayGame()

```
void displayGame (
            const bool game[HEIGHT][WIDTH] )
```

Display the game.

Parameters

game	

2.2.2.4 preFillGame()

```
void preFillGame (
            bool game[HEIGHT][WIDTH] )
```

Index