

labo\_08\_gachet\_jean\_gallay\_david

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	game_of_life.cpp File Reference . . . . .	3
2.1.1	Macro Definition Documentation . . . . .	4
2.1.1.1	print . . . . .	4
2.1.2	Function Documentation . . . . .	4
2.1.2.1	computeMultipleGens() . . . . .	4
2.1.2.2	computeNextGen() . . . . .	4
2.1.2.3	copyArray() . . . . .	5
2.1.2.4	displayGame() . . . . .	5
2.1.2.5	getDisplayChar() . . . . .	5
2.1.2.6	nbOfNeighbours() . . . . .	5
2.1.2.7	preFillGame() . . . . .	6
2.2	game_of_life.h File Reference . . . . .	6
2.2.1	Macro Definition Documentation . . . . .	7
2.2.1.1	ALIVE . . . . .	7
2.2.1.2	ALIVE_CHAR . . . . .	7
2.2.1.3	DEAD . . . . .	7
2.2.1.4	DEAD_CHAR . . . . .	7
2.2.1.5	HEIGHT . . . . .	7
2.2.1.6	WIDTH . . . . .	7
2.2.2	Function Documentation . . . . .	8
2.2.2.1	computeMultipleGens() . . . . .	8
2.2.2.2	computeNextGen() . . . . .	9
2.2.2.3	displayGame() . . . . .	9
2.2.2.4	preFillGame() . . . . .	9
	<b>Index</b>	<b>11</b>



# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">game_of_life.cpp</a>	.....	3
<a href="#">game_of_life.h</a>	.....	6



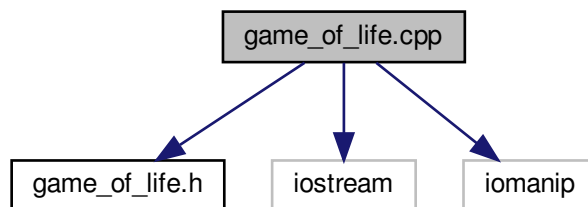
## Chapter 2

# 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

<i>currentGen</i>	[IN] current state of the game [OUT] new state of the game
<i>n</i>	number of generations to compute
<i>autoStop</i>	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

<i>currentGen</i>	[IN] current state of the game [OUT] new state of the game
-------------------	--

#### Returns

if a change occurred



### 2.1.2.3 copyArray()

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

Copys the first array into the second array.

#### Parameters

<i>fromArray</i>	origin array
<i>toArray</i>	target array

### 2.1.2.4 displayGame()

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

Display the game.

#### Parameters

<i>game</i>	
-------------	--

### 2.1.2.5 getDisplayChar()

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

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

#### Parameters

<i>value</i>	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

<i>game</i>	
<i>line</i>	cell's line
<i>column</i>	cell's column

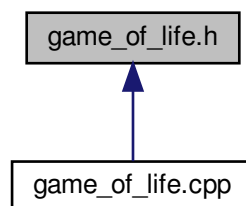
#### Returns

#### 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

<i>currentGen</i>	[IN] current state of the game [OUT] new state of the game
<i>n</i>	number of generations to compute
<i>autoStop</i>	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

<i>currentGen</i>	[IN] current state of the game [OUT] new state of the game
-------------------	--

## Returns

if a change occurred

## 2.2.2.3 displayGame()

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

Display the game.

## Parameters

<i>game</i>	
-------------	--

## 2.2.2.4 preFillGame()

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



# Index

- ALIVE\_CHAR
  - [game\\_of\\_life.h, 7](#)
- ALIVE
  - [game\\_of\\_life.h, 7](#)
- computeMultipleGens
  - [game\\_of\\_life.cpp, 4](#)
  - [game\\_of\\_life.h, 8](#)
- computeNextGen
  - [game\\_of\\_life.cpp, 4](#)
  - [game\\_of\\_life.h, 9](#)
- copyArray
  - [game\\_of\\_life.cpp, 4](#)
- DEAD\_CHAR
  - [game\\_of\\_life.h, 7](#)
- DEAD
  - [game\\_of\\_life.h, 7](#)
- displayGame
  - [game\\_of\\_life.cpp, 5](#)
  - [game\\_of\\_life.h, 9](#)
- game\_of\_life.cpp, 3
  - [computeMultipleGens, 4](#)
  - [computeNextGen, 4](#)
  - [copyArray, 4](#)
  - [displayGame, 5](#)
  - [getDisplayChar, 5](#)
  - [nbOfNeighbours, 5](#)
  - [preFillGame, 6](#)
  - [print, 4](#)
- game\_of\_life.h, 6
  - [ALIVE\\_CHAR, 7](#)
  - [ALIVE, 7](#)
  - [computeMultipleGens, 8](#)
  - [computeNextGen, 9](#)
  - [DEAD\\_CHAR, 7](#)
  - [DEAD, 7](#)
  - [displayGame, 9](#)
  - [HEIGHT, 7](#)
  - [preFillGame, 9](#)
  - [WIDTH, 7](#)
- getDisplayChar
  - [game\\_of\\_life.cpp, 5](#)
- HEIGHT
  - [game\\_of\\_life.h, 7](#)
- nbOfNeighbours
  - [game\\_of\\_life.cpp, 5](#)
- preFillGame
  - [game\\_of\\_life.cpp, 6](#)
  - [game\\_of\\_life.h, 9](#)
- print
  - [game\\_of\\_life.cpp, 4](#)
- WIDTH
  - [game\\_of\\_life.h, 7](#)