Simple language

Generated by Doxygen 1.9.1

Chapter 1

Game of life

Implementacja gry w życie Johna Conwaya

1.0.1 Kod źródłowy

https://github.com/mateuszkojro/ui4_game_of_life

1.0.2 Dokumentacja

https://mateuszkojro.github.io/ui4_game_of_life/

1.1 Wymagania

Aby skompilowac projekt wymagany jest przynajmniej standerd **C++ 14**

1.2 Budowa

Program sklada sie z:

- Interfejs prostego API silnika graficznego (Renderer.h) i jego implementacja w postacji prostego renderera do wyswietlania w konsoli (SimpleConsoleRenderer.h)
- Interfejs prostego API silnika gier (GameEngine.h) i jego implementacja w postaci tematycznej Gry w zycie (GameOfLife.h)

1.3 Zalozenia

W zalozeniu kazda gra stworzona z pomoca API GameEngine i Renderer powinna umozliwiac w prosty sposob zmiane silnika implementacje silnika graficznego moze to byc osiagniete implementujac wszystkie funckcje interfejsu Renderer. A nastepnie przekazujac wskaznik na instancje zaimplementowanego silnika do konfiguracji silnika gry np:

```
GameEngine::Config config;
config.renderer = new SimpleConsoleRenderer;
GameOfLife game_of_life(board, config);
```

gdzie SimpleConsoleRenderer to klasa dziedziczaca po Renderer

1.4 Mozliwa konfiguracja

Istnieje mozliwosc ustawienia poczatkowego stanu planszy za pomocą funkcjonalnosci udostepnionych przez klase `Board`:

1. Zapisywanie i odczytywanie stanu planszy z pliku:

```
    // mozemy otworzyc wczesniej zapisana plansze
const char[] PATH = "saved_board.data";
auto saved_board = Board::load_board(PATH);
// ustawiamy komurke na adresie x=1, y=1 jako aktywna
saved_board(1, 1) = true;
// Zapisujemy wprowadzone dane
Board::save_board(saved_board, "new_board.data")
```

1. Ustawienie zawartosci planszy za pomoca tablicy wartosci boolowskich gdzie `true` znaczy ze komurka bedzie zywa a `false` ze martwa

```
const size_x = 20, size_y = 20;
auto data = new bool[size_x * size_y];
memset(data, true, size_x * size_y);
data[11] = true;
data[12] = true;
data[13] = true;
Board board(data, size_x, size_y);
```

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

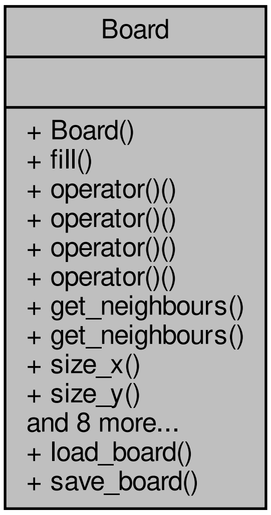
Chapter 5

Class Documentation

5.1 Board Class Reference

```
#include <Board.h>
```

Collaboration diagram for Board:



Public Member Functions

- Board ()=delete
- void fill (bool value)
- bool & operator() (int x, int y)
- bool & operator() (int i)
- bool & operator() (int x, int y) const

    const qualified version operator()(int, int);
- bool & operator() (int i) const

    const qualified version operator()(int);
- std::array< bool, 9 > get_neighbours (int x, int y)
- std::array< bool, 9 > get_neighbours (int i)
- size_t size_x () const

- size_t size_y () const
- bool ∗ get_board () const
- Board (bool ∗board, size_t x, size_t y)
- Board (size_t x, size_t y)
- Board (const Board &other)
- Board (Board &&other) noexcept
- Board & operator= (const Board &)
- unsigned size () const

  Return the size of the underlying array (width ∗ height)
- virtual ∼Board ()

Static Public Member Functions

- static Board load_board (const std::string &path)
- static void save_board (const Board &, const std::string &path)

5.1.1 Detailed Description

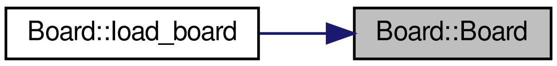
Class containing board (bool array) with functions useful for the implementation of the game of life
Definition at line 14 of file Board.h.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Board() [1/5]

```
Board::Board ( )  [delete]
```

Here is the caller graph for this function:



5.1.2.2 Board() [2/5]

```
Board::Board (
            bool * board,
            size_t x,
            size_t y )
```

Definition at line 9 of file Board.cc.

5.1.2.3 Board() [3/5]

```
Board::Board (
            size_t x,
            size_t y )
```

Definition at line 15 of file Board.cc.

5.1.2.4 Board() [4/5]

```
Board::Board (
              const Board & other )
```

Definition at line 105 of file Board.cc.

5.1.2.5 Board() [5/5]

```
Board::Board (
              Board && other )  [noexcept]
```

Definition at line 116 of file Board.cc.

5.1.2.6 ∼Board()

```
Board::∼Board ( )  [virtual]
```

Definition at line 19 of file Board.cc.

5.1.3 Member Function Documentation

5.1.3.1 fill()

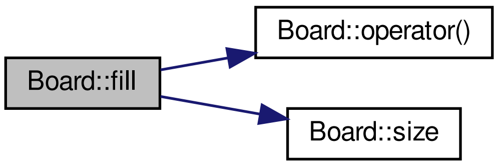
```
void Board::fill (
              bool value )
```

Fils the board wirh specified value

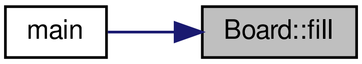
Parameters

value	to fill the buffer with

Definition at line 32 of file Board.cc.
Here is the call graph for this function:



Here is the caller graph for this function:



5.1.3.2 get_board()

```
bool * Board::get_board ( ) const
```

Get the ptr to bool array

Returns

ptr to bool array containing board data

Definition at line 38 of file Board.cc.

5.1.3.3 get_neighbours() [1/2]

```
std::array< bool, 9 > Board::get_neighbours (
              int i )
```

its an analog for get_neighbours(int x, int y); returns array of pairs to neighbours Some might be null

Parameters

i	address of the underlying array

Definition at line 46 of file Board.cc.

5.1.3.4 get_neighbours() [2/2]

```
std::array< bool, 9 > Board::get_neighbours (
              int x,
              int y )
```

Returns array of pairs to neighbours Some might be null

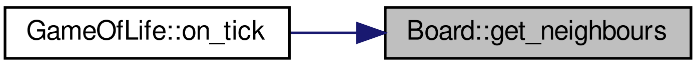
Parameters

x	x coordinate
y	y coordinate

Returns

array of pointers to neighbour cells (some might be null)

Definition at line 42 of file Board.cc.
Here is the caller graph for this function:



5.1.3.5 load_board()

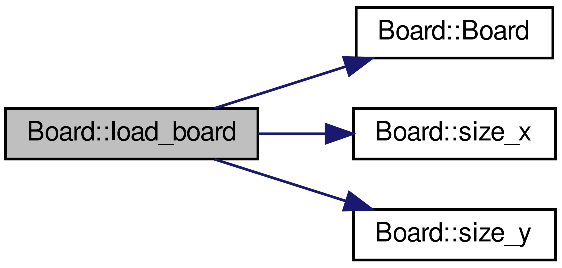
```
Board Board::load_board (
             const std::string & path )  [static]
```

Load board from the file

Parameters

Path	to the file containing the Board

Definition at line 82 of file Board.cc.
Here is the call graph for this function:



5.1.3.6 operator()() [1/4]

```
bool & Board::operator() (
             int i )
```

Accesses element of underlying arr

Parameters

i	Access the ith element of underlying arr

Returns

reference to given field

Definition at line 148 of file Board.cc.

5.1.3.7 operator()() `[2/4]`

```
bool & Board::operator() (
            int i ) const
```

const qualified version operator()(int);
Definition at line 156 of file Board.cc.

5.1.3.8 operator()() `[3/4]`

```
bool & Board::operator() (
            int x,
            int y )
```

Accesses element of underlying arr

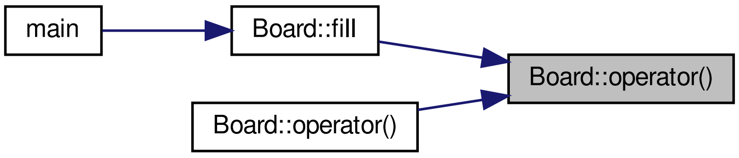
Parameters

x	x coordinate
y	y coordinate

Returns

reference to given field

Definition at line 152 of file Board.cc.
Here is the caller graph for this function:

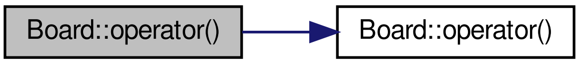


5.1.3.9 operator()() `[4/4]`

```
bool & Board::operator() (
            int x,
            int y ) const
```

const qualified version operator()(int, int);
Definition at line 160 of file Board.cc.

Here is the call graph for this function:



5.1.3.10 operator=()

```
Board & Board::operator= (
             const Board & other )
```

Definition at line 124 of file Board.cc.

5.1.3.11 save_board()

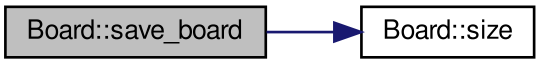
```
void Board::save_board (
             const Board & board,
             const std::string & path )  [static]
```

Save board state to the file

Parameters

Board	to be saved
Path	to the file that bord should be saved to default: "board.save"

Definition at line 95 of file Board.cc.
Here is the call graph for this function:

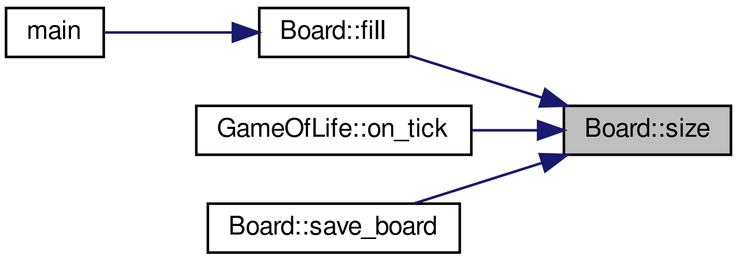


5.1.3.12 size()

```
unsigned Board::size ( ) const
```

Return the size of the underlying array (width $*$ height)
Definition at line 131 of file Board.cc.

Here is the caller graph for this function:



5.1.3.13 size_x()

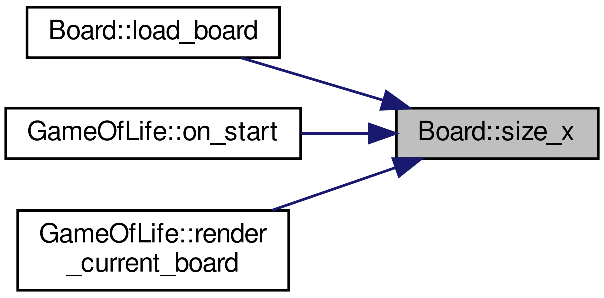
```
size_t Board::size_x ( ) const
```

Return size in x axis

Returns

width of the Board

Definition at line 24 of file Board.cc.
Here is the caller graph for this function:



5.1.3.14 size_y()

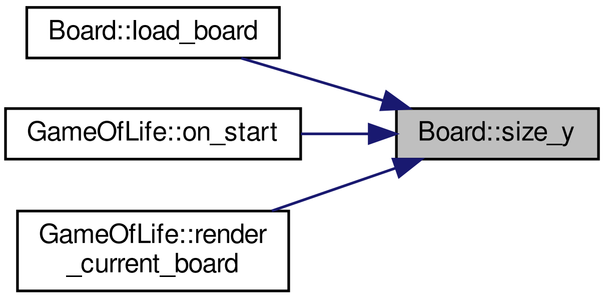
```
size_t Board::size_y ( ) const
```

Return size in y axis

Returns

height in y axis

Definition at line 28 of file Board.cc.
Here is the caller graph for this function:



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

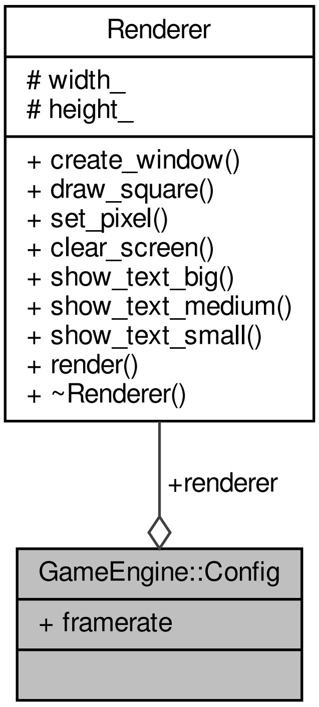
- Board.h
- Board.cc

5.2 GameEngine::Config Struct Reference

Config for game engines.

```
#include <GameEngine.h>
```

Collaboration diagram for GameEngine::Config:



Public Attributes

- int framerate
- Renderer ∗ renderer

5.2.1 Detailed Description

Config for game engines.
Definition at line 25 of file GameEngine.h.

5.2.2 Member Data Documentation

5.2.2.1 framerate

`int GameEngine::Config::framerate`
Definition at line 26 of file GameEngine.h.

5.2.2.2 renderer

`Renderer* GameEngine::Config::renderer`
Definition at line 27 of file GameEngine.h.
The documentation for this struct was generated from the following file:

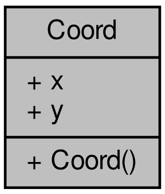
- GameEngine.h

5.3 Coord Struct Reference

Struct containing coordinates of different objects.

`#include <Renderer.h>`

Collaboration diagram for Coord:



Public Member Functions

- Coord (int x_in, int y_in)

Public Attributes

- int x
- int y

5.3.1 Detailed Description

Struct containing coordinates of different objects.
Definition at line 11 of file Renderer.h.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Coord()

```
Coord::Coord (
                int x_in,
                int y_in )  [inline]
```

Definition at line 12 of file Renderer.h.

5.3.3 Member Data Documentation

5.3.3.1 x

```
int Coord::x
```

Definition at line 15 of file Renderer.h.

5.3.3.2 y

`int Coord::y`

Definition at line 16 of file Renderer.h.

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

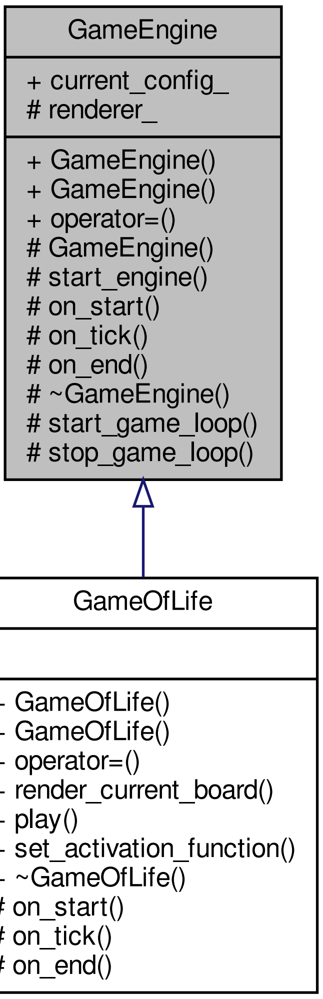
- Renderer.h

5.4 GameEngine Class Reference

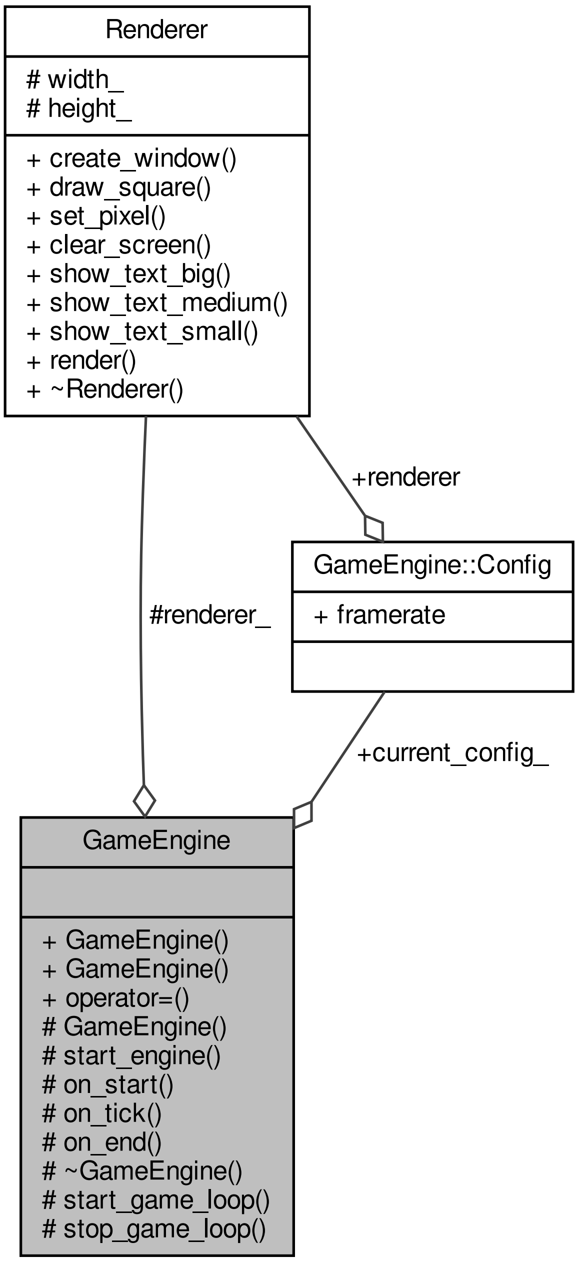
Base class for custom game engines.

`#include <GameEngine.h>`

Inheritance diagram for GameEngine:



Collaboration diagram for GameEngine:



Classes

- struct Config

    Config for game engines.

Public Member Functions

- GameEngine ()=delete

We dont wanna allow creating GameEngine without configuration.
- GameEngine (const GameEngine &)=delete

  We dont wanna allow copying our game engine.
- GameEngine & operator= (const GameEngine &)=delete

Public Attributes

- struct GameEngine::Config current_config_

Protected Member Functions

- GameEngine (const GameEngine::Config &config)

  when we create a GameEngine we always need to give it a config
- virtual void start_engine () final

  game engine will start working
- virtual void on_start ()=0

  This function is called on game start should be overrided by the deriving class.
- virtual void on_tick ()=0

  This function will be invoked on every world tick should be ovverided by the derriving class.
- virtual void on_end ()=0

  This function is called when the game ends should be ovverided by the derriving class.
- virtual ∼GameEngine ()
- void start_game_loop ()

  start main game loop - now every frame on_tick() will be called
- void stop_game_loop ()

  stopping the game loop - the on_end() will be called next

Protected Attributes

- Renderer ∗ renderer_

5.4.1 Detailed Description

Base class for custom game engines.
Definition at line 13 of file GameEngine.h.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 GameEngine() `[1/3]`

```
GameEngine::GameEngine ( )  [delete]
```
We dont wanna allow creating GameEngine without configuration.

5.4.2.2 GameEngine() `[2/3]`

```
GameEngine::GameEngine (
            const GameEngine &  )  [delete]
```
We dont wanna allow copying our game engine.

5.4.2.3 GameEngine() [3/3]

```
GameEngine::GameEngine (
                const GameEngine::Config & config )  [inline], [explicit], [protected]
```

when we create a GameEngine we always need to give it a config
Definition at line 33 of file GameEngine.h.

5.4.2.4 ∼GameEngine()

```
virtual GameEngine::∼GameEngine ( )  [inline], [protected], [virtual]
```

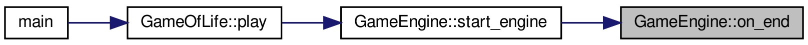
Definition at line 66 of file GameEngine.h.

5.4.3 Member Function Documentation

5.4.3.1 on_end()

```
virtual void GameEngine::on_end ( )  [protected], [pure virtual]
```

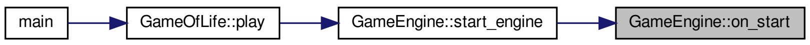
This function is called when the game ends should be ovverided by the derriving class.
Implemented in GameOfLife.
Here is the caller graph for this function:



5.4.3.2 on_start()

```
virtual void GameEngine::on_start ( )  [protected], [pure virtual]
```

This function is called on game start should be overrided by the deriving class.
Implemented in GameOfLife.
Here is the caller graph for this function:

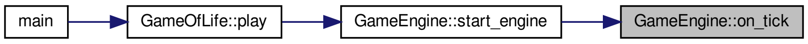


5.4.3.3 on_tick()

```
virtual void GameEngine::on_tick ( )  [protected], [pure virtual]
```

This function will be invoked on every world tick should be ovverided by the derriving class.
Implemented in GameOfLife.

Here is the caller graph for this function:



5.4.3.4 operator=()

```
GameEngine& GameEngine::operator= (
             const GameEngine & )  [delete]
```

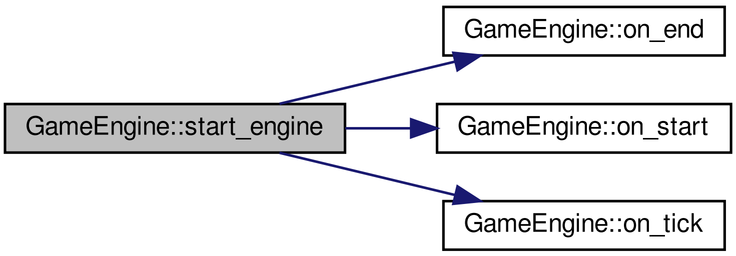
5.4.3.5 start_engine()

```
virtual void GameEngine::start_engine ( )  [inline], [final], [protected], [virtual]
```

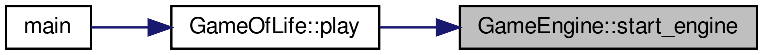
game engine will start working

Definition at line 39 of file GameEngine.h.

Here is the call graph for this function:



Here is the caller graph for this function:



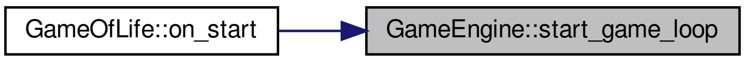
5.4.3.6 start_game_loop()

```
void GameEngine::start_game_loop ( )  [inline], [protected]
```

start main game loop - now every frame on_tick() will be called

Definition at line 70 of file GameEngine.h.

Here is the caller graph for this function:



5.4.3.7 stop_game_loop()

`void GameEngine::stop_game_loop ( )  [inline], [protected]`

stopping the game loop - the on_end() will be called next
Definition at line 73 of file GameEngine.h.

5.4.4 Member Data Documentation

5.4.4.1 current_config_

`struct GameEngine::Config GameEngine::current_config_`

5.4.4.2 renderer_

`Renderer* GameEngine::renderer_  [protected]`

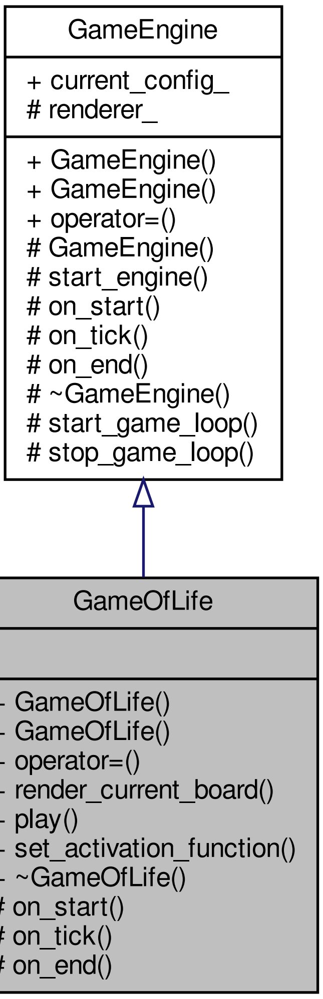
Definition at line 75 of file GameEngine.h.
The documentation for this class was generated from the following file:

- GameEngine.h

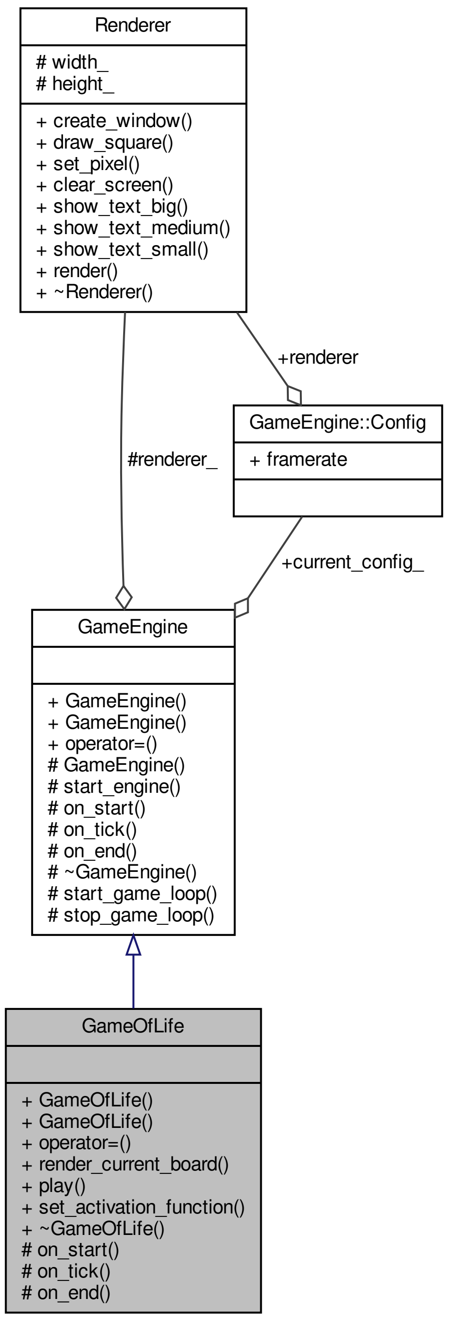
5.5 GameOfLife Class Reference

Implementation of the game of life.
`#include <GameOfLife.h>`

Inheritance diagram for GameOfLife:



Collaboration diagram for GameOfLife:



Public Member Functions

- GameOfLife (const Board &board, const Config &config)
- GameOfLife (const GameOfLife &)=delete
- const GameOfLife & operator= (const GameOfLife &)=delete
- void render_current_board ()

    *render current_board_*
- void play ()

start the game engine
- void set_activation_function (bool(∗func)(bool, int))
- ∼GameOfLife () override

Protected Member Functions

- void on_start () override

  This function is called on game start should be overrided by the deriving class.
- void on_tick () override

  This function will be invoked on every world tick should be ovverided by the derriving class.
- void on_end () override

  This function is called when the game ends should be ovverided by the derriving class.

Additional Inherited Members

5.5.1 Detailed Description

Implementation of the game of life.
Definition at line 18 of file GameOfLife.h.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 GameOfLife() `[1/2]`

```
GameOfLife::GameOfLife (
              const Board & board,
              const Config & config )  [inline], [explicit]
```
Definition at line 20 of file GameOfLife.h.

5.5.2.2 GameOfLife() `[2/2]`

```
GameOfLife::GameOfLife (
              const GameOfLife & )  [delete]
```

5.5.2.3 ∼GameOfLife()

```
GameOfLife::∼GameOfLife ( )  [override]
```
sets the function that will be used to determine if cell should be alive

Parameters

func	returning bool (if is alive) has params (bool) is currently alive (int) how many neighbours it has

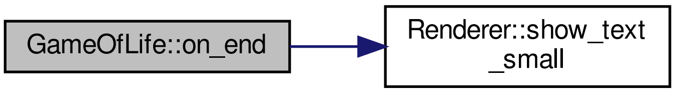
Definition at line 26 of file GameOfLife.cc.

5.5.3 Member Function Documentation

5.5.3.1 on_end()

```
void GameOfLife::on_end ( )  [override], [protected], [virtual]
```
This function is called when the game ends should be ovverided by the derriving class.
Implements GameEngine.

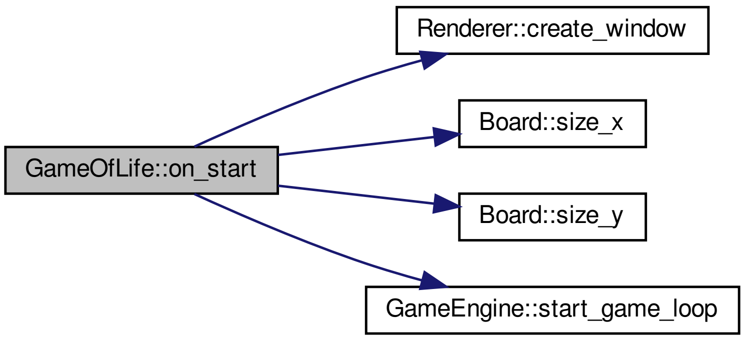
Definition at line 103 of file GameOfLife.cc.
Here is the call graph for this function:



5.5.3.2 on_start()

```
void GameOfLife::on_start ( )  [override], [protected], [virtual]
```

This function is called on game start should be overrided by the deriving class.
Implements GameEngine.
Definition at line 33 of file GameOfLife.cc.
Here is the call graph for this function:

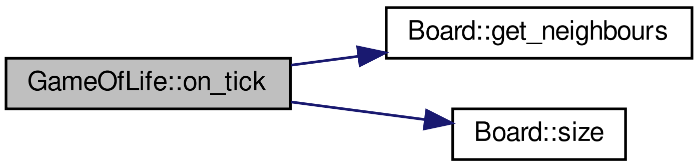


5.5.3.3 on_tick()

```
void GameOfLife::on_tick ( )  [override], [protected], [virtual]
```

This function will be invoked on every world tick should be ovverided by the derriving class.
Implements GameEngine.
Definition at line 65 of file GameOfLife.cc.

Here is the call graph for this function:



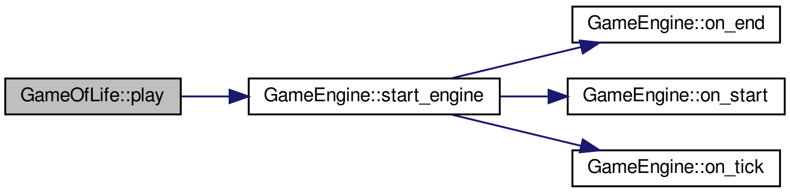
5.5.3.4 operator=()

```
const GameOfLife& GameOfLife::operator= (
             const GameOfLife & )  [delete]
```

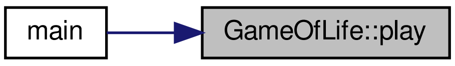
5.5.3.5 play()

```
void GameOfLife::play ( )
```

start the game engine
Definition at line 18 of file GameOfLife.cc.
Here is the call graph for this function:



Here is the caller graph for this function:



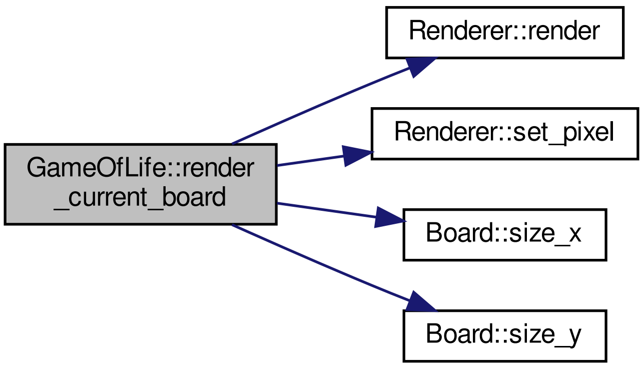
5.5.3.6 render_current_board()

```
void GameOfLife::render_current_board ( )
```

render current_board_

Definition at line 50 of file GameOfLife.cc.

Here is the call graph for this function:



5.5.3.7 set_activation_function()

```
void GameOfLife::set_activation_function (
                bool(*)(bool, int) func )
```

Sets the function that will be used to determine if the cell should be alive

Parameters

func	function ptr function should return bool (true if a cell should be alive) based on the number of alive neighbours and if given cell is alive

Definition at line 22 of file GameOfLife.cc.

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

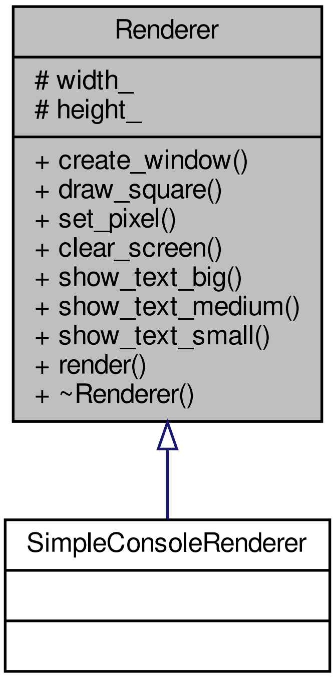
- GameOfLife.h
- GameOfLife.cc

5.6 Renderer Class Reference

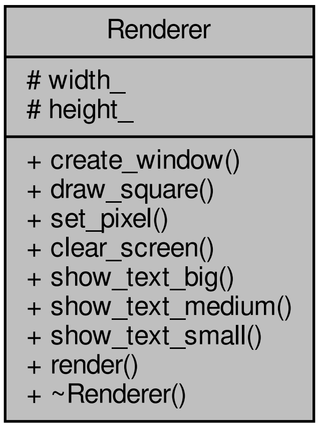
Basic base class for all renderers.

```
#include <Renderer.h>
```

Inheritance diagram for Renderer:



Collaboration diagram for Renderer:



Public Member Functions

- virtual void create_window (int size_x, int size_y)=0

Creates window of given size.
- virtual void draw_square (const Coord &position, int size_x, int size_y, const Color &fill)=0

  Draws a square on position wit size and fill.
- virtual void set_pixel (const Coord &position, const Color &fill)=0
- virtual void clear_screen (const Color &fill)=0

  Fill all screen with defined color.
- virtual void show_text_big (const Coord &position, const std::string &text)=0
- virtual void show_text_medium (const Coord &position, const std::string &text)=0
- virtual void show_text_small (const Coord &position, const std::string &text)=0
- virtual void render ()=0
- virtual ∼Renderer ()=default

Protected Attributes

- int width_

  width of the render plane
- int height_

  height of the render plane

5.6.1 Detailed Description

Basic base class for all renderers.
Definition at line 29 of file Renderer.h.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 ∼Renderer()

```
virtual Renderer::∼Renderer ( )  [virtual], [default]
```

5.6.3 Member Function Documentation

5.6.3.1 clear_screen()

```
virtual void Renderer::clear_screen (
             const Color & fill )  [pure virtual]
```

Fill all screen with defined color.

Parameters

Color	color to fill the screen with

5.6.3.2 create_window()

```
virtual void Renderer::create_window (
             int size_x,
             int size_y )  [pure virtual]
```

Creates window of given size.

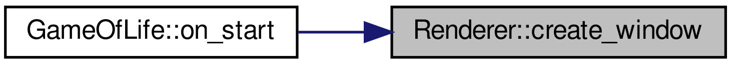
Parameters

int	Size in x dimension

Parameters

int	Size in y dimension

Here is the caller graph for this function:



5.6.3.3 draw_square()

```
virtual void Renderer::draw_square (
             const Coord & position,
             int size_x,
             int size_y,
             const Color & fill )  [pure virtual]
```

Draws a square on position wit size and fill.

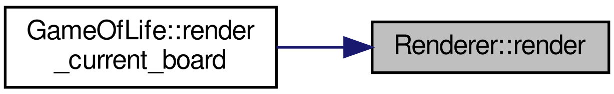
Parameters

Coord	position
size	in x axis
size	in y axis
Color	color to fill square with

5.6.3.4 render()

```
virtual void Renderer::render ( )  [pure virtual]
```

Here is the caller graph for this function:



5.6.3.5 set_pixel()

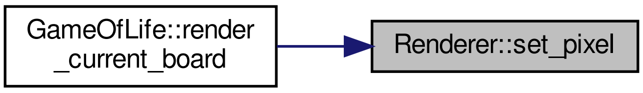
```
virtual void Renderer::set_pixel (
            const Coord & position,
            const Color & fill )  [pure virtual]
```

Sets the position at given coordinate to given Color

Parameters

position	const Coord& position of the pixels to be set
fill	The color to set the pixel to

Here is the caller graph for this function:



5.6.3.6 show_text_big()

```
virtual void Renderer::show_text_big (
            const Coord & position,
            const std::string & text )  [pure virtual]
```

Show text in big letters on position

Parameters

position	Coord of the beginning of the text
text	text to be printed

5.6.3.7 show_text_medium()

```
virtual void Renderer::show_text_medium (
            const Coord & position,
            const std::string & text )  [pure virtual]
```

Show text in medium letters on position

Parameters

position	Coord of the beginning of the text
text	text to be printed

5.6.3.8 show_text_small()

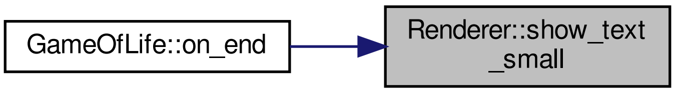
```
virtual void Renderer::show_text_small (
            const Coord & position,
            const std::string & text )  [pure virtual]
```

Show text in small letters on position

Parameters

position	Coord of the beginning of the text
text	text to be printed

Here is the caller graph for this function:



5.6.4 Member Data Documentation

5.6.4.1 height_

`int Renderer::height_ [protected]`
height of the render plane
Definition at line 76 of file Renderer.h.

5.6.4.2 width_

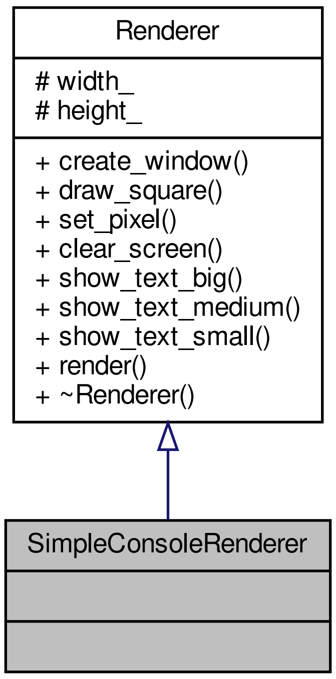
`int Renderer::width_ [protected]`
width of the render plane
Definition at line 74 of file Renderer.h.
The documentation for this class was generated from the following file:

- Renderer.h

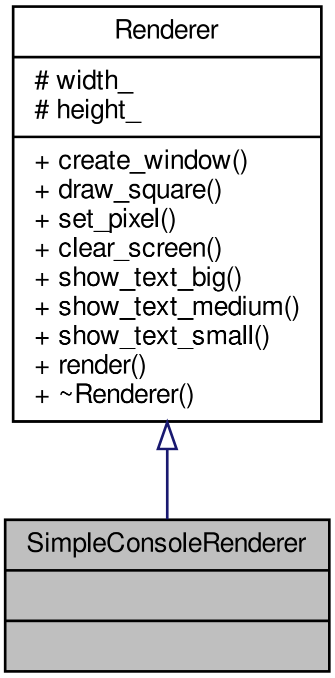
5.7 SimpleConsoleRenderer Class Reference

```
#include <SimpleConsoleRenderer.h>
```

Inheritance diagram for SimpleConsoleRenderer:



Collaboration diagram for SimpleConsoleRenderer:



Additional Inherited Members

5.7.1 Detailed Description

Definition at line 12 of file SimpleConsoleRenderer.h.
The documentation for this class was generated from the following files:

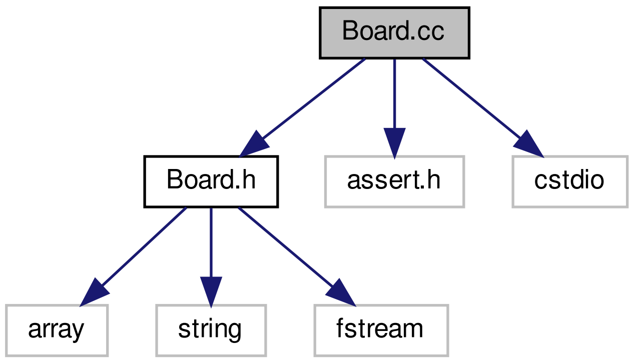
- SimpleConsoleRenderer.h
- SimpleConsoleRenderer.cc

Chapter 6

File Documentation

6.1 Board.cc File Reference

```
#include "Board.h"
#include <assert.h>
#include <cstdio>
```
Include dependency graph for Board.cc:



6.2 Board.cc

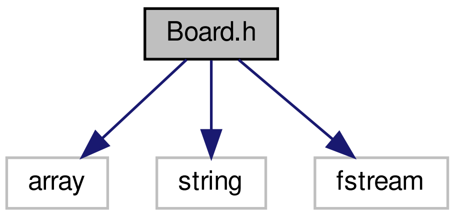
```
//
// Created by mateu on 4/1/2021.
//

#include "Board.h"
#include <assert.h>
#include <cstdio>

Board::Board(bool *board, size_t x, size_t y) :
        size_x_(x),
        size_y_(y) {
    board_ = board;
}

Board::Board(size_t x, size_t y) : size_x_(x), size_y_(y) {
  board_ = new bool[x * y];
}

Board::~Board() {
    delete[] board_;
    board_ = nullptr;
}

size_t Board::size_x() const {
    return size_x_;
}

size_t Board::size_y() const {
    return size_y_;
}

void Board::fill(bool value) {
  for (int i = 0; i < size(); i++) {
    operator()(i) = value;
  }
}

bool *Board::get_board() const {
    return board_;
}

std::array<bool, 9> Board::get_neighbours(int x, int y) {
    return get_neighbours(translate_adress(x, y));
}

std::array<bool, 9> Board::get_neighbours(int i) {
    std::array<bool, 9> result{};


    int pos_x, pos_y;

    // Transform continuous address to x, y one
    pos_y = i / size_x_;
    pos_x = i % size_x_;

    // place in out array
    int itr = 0;


    for (int y = -1; y <= 1; y++) {
        for (int x = -1; x <= 1; x++) {

            // transform to position on the board
            int board_x = pos_x + x;
            int board_y = pos_y + y;
            int board_i = translate_adress(board_x, board_y);

            bool is_target = (board_x == pos_x) && (board_y == pos_y);
            bool is_valid = (board_i != -1);

            if (!is_target && is_valid){
                result[itr++] = board_[board_i];
            } else {
                result[itr++] = false;
            }
        }
    }

    return result;
}

Board Board::load_board(const std::string &path) {
    std::fstream file;
    file.open(path, std::ios::in);
    size_t size_x, size_y;

    file » size_x » size_y;
    bool *board = new bool[size_y * size_x];
    for (int i = 0; i < size_x * size_y; ++i) {
        file » board[i];
    }
    return Board(board, size_x, size_y);
}

void Board::save_board(const Board &board, const std::string &path) {
    std::fstream file;
    file.open(path, std::ios::out);
    file « board.size_x_;
    file « board.size_y_;
    for (int i = 0; i < board.size(); ++i) {
        file « board(i);
    }
}

Board::Board(const Board &other) {
    copy(other);
}

```

```
void Board::copy(const Board &other) {
    size_y_ = other.size_y_;
    size_x_ = other.size_x_;
    board_ = new bool[size()];
    memcpy(board_, other.board_, size());
}

Board::Board(Board &&other) noexcept {
    assert(false);
    size_y_ = other.size_y_;
    size_x_ = other.size_x_;
    board_ = other.board_;
    other.board_ = nullptr;
}

Board &Board::operator=(const Board &other) {
    if (this == &other)
        return *this;
    copy(other);
    return *this;
}

unsigned Board::size() const {
    return size_x_ * size_y_;
}

int Board::translate_adress(int x, int y) const {
  if (y >= size_y_)
    return -1;
  if (x >= size_x_)
    return -1;
  if (x < 0)
    return -1;
  if (y < 0)
    return -1;

  return y * size_x_ + x;
}

bool &Board::operator()(int i) {
    return board_[i];
}

bool &Board::operator()(int x, int y) {
    return operator()(translate_adress(x, y));
}

bool &Board::operator()(int i) const {
    return board_[i];
}

bool &Board::operator()(int x, int y) const {
    return operator()(translate_adress(x, y));
}
```

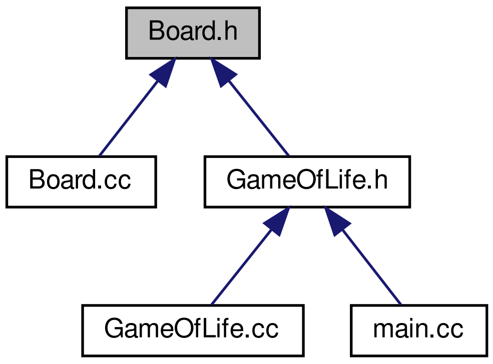
6.3 Board.h File Reference

```
#include <array>
#include <string>
#include <fstream>
```

Include dependency graph for Board.h:



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



Classes

- class Board

6.4 Board.h

```
//
// Created by mateu on 4/1/2021.
//

#ifndef GAME_OF_LIFE_BOARD_H
#define GAME_OF_LIFE_BOARD_H

#include <array>
#include <string>
#include <fstream>

class Board {
public:

    Board() = delete;

    void fill(bool value);

    bool &operator()(int x, int y);

    bool &operator()(int i);

    bool &operator()(int x, int y) const;

    bool &operator()(int i) const;

    std::array<bool, 9> get_neighbours(int x, int y);

    std::array<bool, 9> get_neighbours(int i);

    static Board load_board(const std::string &path);

    static void save_board(const Board &, const std::string &path);

    size_t size_x() const;

    size_t size_y() const;

    bool *get_board() const;

public:


    Board(bool *board, size_t x, size_t y);

    Board(size_t x, size_t y);

    Board(const Board &other);

    Board(Board &&other) noexcept;

    Board &operator=(const Board &);

    unsigned size() const;

    virtual ~Board();


private:
    int translate_adress(int x, int y) const;

    void copy(const Board &other);

    size_t size_x_;
    size_t size_y_;

    bool *board_;

};


#endif //GAME_OF_LIFE_BOARD_H
```

6.5 build/CMakeFiles/3.19.7/CompilerIdC/CMakeCCompilerId.c File Reference

Macros

- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X) #X
- #define STRINGIFY(X) STRINGIFY_HELPER(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define C_DIALECT

Functions

- int main (int argc, char *argv[ ])

Variables

- char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"

- char const ∗ info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const ∗ info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char ∗ info_language_dialect_default

6.5.1 Macro Definition Documentation

6.5.1.1 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```
Definition at line 561 of file CMakeCCompilerId.c.

6.5.1.2 C_DIALECT

```
#define C_DIALECT
```
Definition at line 645 of file CMakeCCompilerId.c.

6.5.1.3 COMPILER_ID

```
#define COMPILER_ID ""
```
Definition at line 314 of file CMakeCCompilerId.c.

6.5.1.4 DEC

```
#define DEC(
                n )
```
Value:
```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```
Definition at line 565 of file CMakeCCompilerId.c.

6.5.1.5 HEX

```
#define HEX(
                n )
```
Value:
```
  ('0' + ((n)»28 & 0xF)), \
  ('0' + ((n)»24 & 0xF)), \
  ('0' + ((n)»20 & 0xF)), \
  ('0' + ((n)»16 & 0xF)), \
  ('0' + ((n)»12 & 0xF)), \
  ('0' + ((n)»8  & 0xF)), \
  ('0' + ((n)»4  & 0xF)), \
  ('0' + ((n)     & 0xF))
```
Definition at line 576 of file CMakeCCompilerId.c.

6.5.1.6 PLATFORM_ID

```
#define PLATFORM_ID
```
Definition at line 439 of file CMakeCCompilerId.c.

6.5.1.7 STRINGIFY

```
#define STRINGIFY(
              X ) STRINGIFY_HELPER(X)
```
Definition at line 335 of file CMakeCCompilerId.c.

6.5.1.8 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
              X ) #X
```
Definition at line 334 of file CMakeCCompilerId.c.

6.5.2 Function Documentation

6.5.2.1 main()

```
int main (
             int argc,
             char * argv[] )
```
Definition at line 665 of file CMakeCCompilerId.c.

6.5.3 Variable Documentation

6.5.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```
Definition at line 636 of file CMakeCCompilerId.c.

6.5.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```
Definition at line 321 of file CMakeCCompilerId.c.

6.5.3.3 info_language_dialect_default

```
const char* info_language_dialect_default
```
Initial value:
```
=
  "INFO" ":" "dialect_default[" C_DIALECT "]"
```
Definition at line 654 of file CMakeCCompilerId.c.

6.5.3.4 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```
Definition at line 635 of file CMakeCCompilerId.c.

6.6 CMakeCCompilerId.c

```
#ifdef __cplusplus
# error "A C++ compiler has been selected for C."
#endif

#if defined(__18CXX)
```

```
# define ID_VOID_MAIN
#endif
#if defined(__CLASSIC_C__)
/* cv-qualifiers did not exist in K&R C */
# define const
# define volatile
#endif


/* Version number components: V=Version, R=Revision, P=Patch
   Version date components:   YYYY=Year, MM=Month,   DD=Day  */

#if defined(__INTEL_COMPILER) || defined(__ICC)
# define COMPILER_ID "Intel"
# if defined(_MSC_VER)
#  define SIMULATE_ID "MSVC"
# endif
# if defined(__GNUC__)
#  define SIMULATE_ID "GNU"
# endif
  /* __INTEL_COMPILER = VRP */
# define COMPILER_VERSION_MAJOR DEC(__INTEL_COMPILER/100)
# define COMPILER_VERSION_MINOR DEC(__INTEL_COMPILER/10 % 10)
# if defined(__INTEL_COMPILER_UPDATE)
#  define COMPILER_VERSION_PATCH DEC(__INTEL_COMPILER_UPDATE)
# else
#  define COMPILER_VERSION_PATCH DEC(__INTEL_COMPILER   % 10)
# endif
# if defined(__INTEL_COMPILER_BUILD_DATE)
  /* __INTEL_COMPILER_BUILD_DATE = YYYYMMDD */
#  define COMPILER_VERSION_TWEAK DEC(__INTEL_COMPILER_BUILD_DATE)
# endif
# if defined(_MSC_VER)
   /* _MSC_VER = VVRR */
#  define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
#  define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
# endif
# if defined(__GNUC__)
#  define SIMULATE_VERSION_MAJOR DEC(__GNUC__)
# elif defined(__GNUG__)
#  define SIMULATE_VERSION_MAJOR DEC(__GNUG__)
# endif
# if defined(__GNUC_MINOR__)
#  define SIMULATE_VERSION_MINOR DEC(__GNUC_MINOR__)
# endif
# if defined(__GNUC_PATCHLEVEL__)
#  define SIMULATE_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
# endif

#elif defined(__PATHCC__)
# define COMPILER_ID "PathScale"
# define COMPILER_VERSION_MAJOR DEC(__PATHCC__)
# define COMPILER_VERSION_MINOR DEC(__PATHCC_MINOR__)
# if defined(__PATHCC_PATCHLEVEL__)
#  define COMPILER_VERSION_PATCH DEC(__PATHCC_PATCHLEVEL__)
# endif

#elif defined(__BORLANDC__) && defined(__CODEGEARC_VERSION__)
# define COMPILER_ID "Embarcadero"
# define COMPILER_VERSION_MAJOR HEX(__CODEGEARC_VERSION__»24 & 0x00FF)
# define COMPILER_VERSION_MINOR HEX(__CODEGEARC_VERSION__»16 & 0x00FF)
# define COMPILER_VERSION_PATCH DEC(__CODEGEARC_VERSION__     & 0xFFFF)

#elif defined(__BORLANDC__)
# define COMPILER_ID "Borland"
  /* __BORLANDC__ = 0xVRR */
# define COMPILER_VERSION_MAJOR HEX(__BORLANDC__»8)
# define COMPILER_VERSION_MINOR HEX(__BORLANDC__ & 0xFF)

#elif defined(__WATCOMC__) && __WATCOMC__ < 1200
# define COMPILER_ID "Watcom"
   /* __WATCOMC__ = VVRR */
# define COMPILER_VERSION_MAJOR DEC(__WATCOMC__ / 100)
# define COMPILER_VERSION_MINOR DEC((__WATCOMC__ / 10) % 10)
# if (__WATCOMC__ % 10) > 0
#  define COMPILER_VERSION_PATCH DEC(__WATCOMC__ % 10)
# endif

#elif defined(__WATCOMC__)
# define COMPILER_ID "OpenWatcom"
   /* __WATCOMC__ = VVRP + 1100 */
# define COMPILER_VERSION_MAJOR DEC((__WATCOMC__ - 1100) / 100)
# define COMPILER_VERSION_MINOR DEC((__WATCOMC__ / 10) % 10)
# if (__WATCOMC__ % 10) > 0
#  define COMPILER_VERSION_PATCH DEC(__WATCOMC__ % 10)
# endif

```

```
#elif defined(__SUNPRO_C)
# define COMPILER_ID "SunPro"
# if __SUNPRO_C >= 0x5100
   /* __SUNPRO_C = 0xVRRP */
#  define COMPILER_VERSION_MAJOR HEX(__SUNPRO_C»12)
#  define COMPILER_VERSION_MINOR HEX(__SUNPRO_C»4 & 0xFF)
#  define COMPILER_VERSION_PATCH HEX(__SUNPRO_C    & 0xF)
# else
   /* __SUNPRO_CC = 0xVRP */
#  define COMPILER_VERSION_MAJOR HEX(__SUNPRO_C»8)
#  define COMPILER_VERSION_MINOR HEX(__SUNPRO_C»4 & 0xF)
#  define COMPILER_VERSION_PATCH HEX(__SUNPRO_C    & 0xF)
# endif

#elif defined(__HP_cc)
# define COMPILER_ID "HP"
  /* __HP_cc = VVRRPP */
# define COMPILER_VERSION_MAJOR DEC(__HP_cc/10000)
# define COMPILER_VERSION_MINOR DEC(__HP_cc/100 % 100)
# define COMPILER_VERSION_PATCH DEC(__HP_cc     % 100)

#elif defined(__DECC)
# define COMPILER_ID "Compaq"
  /* __DECC_VER = VVRRTPPPP */
# define COMPILER_VERSION_MAJOR DEC(__DECC_VER/10000000)
# define COMPILER_VERSION_MINOR DEC(__DECC_VER/100000  % 100)
# define COMPILER_VERSION_PATCH DEC(__DECC_VER         % 10000)

#elif defined(__IBMC__) && defined(__COMPILER_VER__)
# define COMPILER_ID "zOS"
  /* __IBMC__ = VRP */
# define COMPILER_VERSION_MAJOR DEC(__IBMC__/100)
# define COMPILER_VERSION_MINOR DEC(__IBMC__/10 % 10)
# define COMPILER_VERSION_PATCH DEC(__IBMC__    % 10)

#elif defined(__ibmxl__) && defined(__clang__)
# define COMPILER_ID "XLClang"
# define COMPILER_VERSION_MAJOR DEC(__ibmxl_version__)
# define COMPILER_VERSION_MINOR DEC(__ibmxl_release__)
# define COMPILER_VERSION_PATCH DEC(__ibmxl_modification__)
# define COMPILER_VERSION_TWEAK DEC(__ibmxl_ptf_fix_level__)


#elif defined(__IBMC__) && !defined(__COMPILER_VER__) && __IBMC__ >= 800
# define COMPILER_ID "XL"
  /* __IBMC__ = VRP */
# define COMPILER_VERSION_MAJOR DEC(__IBMC__/100)
# define COMPILER_VERSION_MINOR DEC(__IBMC__/10 % 10)
# define COMPILER_VERSION_PATCH DEC(__IBMC__    % 10)

#elif defined(__IBMC__) && !defined(__COMPILER_VER__) && __IBMC__ < 800
# define COMPILER_ID "VisualAge"
  /* __IBMC__ = VRP */
# define COMPILER_VERSION_MAJOR DEC(__IBMC__/100)
# define COMPILER_VERSION_MINOR DEC(__IBMC__/10 % 10)
# define COMPILER_VERSION_PATCH DEC(__IBMC__    % 10)

#elif defined(__PGI)
# define COMPILER_ID "PGI"
# define COMPILER_VERSION_MAJOR DEC(__PGIC__)
# define COMPILER_VERSION_MINOR DEC(__PGIC_MINOR__)
# if defined(__PGIC_PATCHLEVEL__)
#  define COMPILER_VERSION_PATCH DEC(__PGIC_PATCHLEVEL__)
# endif

#elif defined(_CRAYC)
# define COMPILER_ID "Cray"
# define COMPILER_VERSION_MAJOR DEC(_RELEASE_MAJOR)
# define COMPILER_VERSION_MINOR DEC(_RELEASE_MINOR)

#elif defined(__TI_COMPILER_VERSION__)
# define COMPILER_ID "TI"
  /* __TI_COMPILER_VERSION__ = VVVRRRPPP */
# define COMPILER_VERSION_MAJOR DEC(__TI_COMPILER_VERSION__/1000000)
# define COMPILER_VERSION_MINOR DEC(__TI_COMPILER_VERSION__/1000   % 1000)
# define COMPILER_VERSION_PATCH DEC(__TI_COMPILER_VERSION__        % 1000)

#elif defined(__FUJITSU) || defined(__FCC_VERSION) || defined(__fcc_version)
# define COMPILER_ID "Fujitsu"

#elif defined(__ghs__)
# define COMPILER_ID "GHS"
/* __GHS_VERSION_NUMBER = VVVVRP */
# ifdef __GHS_VERSION_NUMBER
# define COMPILER_VERSION_MAJOR DEC(__GHS_VERSION_NUMBER / 100)
# define COMPILER_VERSION_MINOR DEC(__GHS_VERSION_NUMBER / 10 % 10)
# define COMPILER_VERSION_PATCH DEC(__GHS_VERSION_NUMBER      % 10)
# endif

#elif defined(__TINYC__)
# define COMPILER_ID "TinyCC"

#elif defined(__BCC__)
# define COMPILER_ID "Bruce"

#elif defined(__SCO_VERSION__)
# define COMPILER_ID "SCO"

#elif defined(__ARMCC_VERSION) && !defined(__clang__)
# define COMPILER_ID "ARMCC"
#if __ARMCC_VERSION >= 1000000
  /* __ARMCC_VERSION = VRRPPPP */
  # define COMPILER_VERSION_MAJOR DEC(__ARMCC_VERSION/1000000)
  # define COMPILER_VERSION_MINOR DEC(__ARMCC_VERSION/10000 % 100)
  # define COMPILER_VERSION_PATCH DEC(__ARMCC_VERSION     % 10000)
#else
  /* __ARMCC_VERSION = VRPPPP */
  # define COMPILER_VERSION_MAJOR DEC(__ARMCC_VERSION/100000)
  # define COMPILER_VERSION_MINOR DEC(__ARMCC_VERSION/10000 % 10)
  # define COMPILER_VERSION_PATCH DEC(__ARMCC_VERSION    % 10000)
#endif


#elif defined(__clang__) && defined(__apple_build_version__)
# define COMPILER_ID "AppleClang"
# if defined(_MSC_VER)
#  define SIMULATE_ID "MSVC"
# endif
# define COMPILER_VERSION_MAJOR DEC(__clang_major__)
# define COMPILER_VERSION_MINOR DEC(__clang_minor__)
# define COMPILER_VERSION_PATCH DEC(__clang_patchlevel__)
# if defined(_MSC_VER)
   /* _MSC_VER = VVRR */
#  define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
#  define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
# endif
# define COMPILER_VERSION_TWEAK DEC(__apple_build_version__)

#elif defined(__clang__) && defined(__ARMCOMPILER_VERSION)
# define COMPILER_ID "ARMClang"
  # define COMPILER_VERSION_MAJOR DEC(__ARMCOMPILER_VERSION/1000000)
  # define COMPILER_VERSION_MINOR DEC(__ARMCOMPILER_VERSION/10000 % 100)
  # define COMPILER_VERSION_PATCH DEC(__ARMCOMPILER_VERSION     % 10000)
# define COMPILER_VERSION_INTERNAL DEC(__ARMCOMPILER_VERSION)

#elif defined(__clang__)
# define COMPILER_ID "Clang"
# if defined(_MSC_VER)
#  define SIMULATE_ID "MSVC"
# endif
# define COMPILER_VERSION_MAJOR DEC(__clang_major__)
# define COMPILER_VERSION_MINOR DEC(__clang_minor__)
# define COMPILER_VERSION_PATCH DEC(__clang_patchlevel__)
# if defined(_MSC_VER)
   /* _MSC_VER = VVRR */
#  define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
#  define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
# endif

#elif defined(__GNUC__)
# define COMPILER_ID "GNU"
# define COMPILER_VERSION_MAJOR DEC(__GNUC__)
# if defined(__GNUC_MINOR__)
#  define COMPILER_VERSION_MINOR DEC(__GNUC_MINOR__)
# endif
# if defined(__GNUC_PATCHLEVEL__)
#  define COMPILER_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
# endif

#elif defined(_MSC_VER)
# define COMPILER_ID "MSVC"
  /* _MSC_VER = VVRR */
# define COMPILER_VERSION_MAJOR DEC(_MSC_VER / 100)
# define COMPILER_VERSION_MINOR DEC(_MSC_VER % 100)
# if defined(_MSC_FULL_VER)
#  if _MSC_VER >= 1400
    /* _MSC_FULL_VER = VVRRPPPPP */
#   define COMPILER_VERSION_PATCH DEC(_MSC_FULL_VER % 100000)
#  else
    /* _MSC_FULL_VER = VVRRPPPP */
#   define COMPILER_VERSION_PATCH DEC(_MSC_FULL_VER % 10000)
#  endif
# endif
# if defined(_MSC_BUILD)
#  define COMPILER_VERSION_TWEAK DEC(_MSC_BUILD)
# endif

#elif defined(__VISUALDSPVERSION__) || defined(__ADSPBLACKFIN__) || defined(__ADSPTS__) || defined(__ADSP21000__)
# define COMPILER_ID "ADSP"
#if defined(__VISUALDSPVERSION__)
  /* __VISUALDSPVERSION__ = 0xVVRRPP00 */
# define COMPILER_VERSION_MAJOR HEX(__VISUALDSPVERSION__»24)
# define COMPILER_VERSION_MINOR HEX(__VISUALDSPVERSION__»16 & 0xFF)
# define COMPILER_VERSION_PATCH HEX(__VISUALDSPVERSION__»8  & 0xFF)
#endif

#elif defined(__IAR_SYSTEMS_ICC__) || defined(__IAR_SYSTEMS_ICC)
# define COMPILER_ID "IAR"
# if defined(__VER__) && defined(__ICCARM__)
#  define COMPILER_VERSION_MAJOR DEC((__VER__) / 1000000)
#  define COMPILER_VERSION_MINOR DEC(((__VER__) / 1000) % 1000)
#  define COMPILER_VERSION_PATCH DEC((__VER__) % 1000)
#  define COMPILER_VERSION_INTERNAL DEC(__IAR_SYSTEMS_ICC__)
# elif defined(__VER__) && (defined(__ICCAVR__) || defined(__ICCRX__) || defined(__ICCRH850__) || defined(__ICCRL78__) || defined(__ICC430__) || defined(__ICCRISCV__) || defined(__ICCV850__) || defined(__ICC8051__))
#  define COMPILER_VERSION_MAJOR DEC((__VER__) / 100)
#  define COMPILER_VERSION_MINOR DEC((__VER__) - (((__VER__) / 100)*100))
#  define COMPILER_VERSION_PATCH DEC(__SUBVERSION__)
#  define COMPILER_VERSION_INTERNAL DEC(__IAR_SYSTEMS_ICC__)
# endif

#elif defined(__SDCC_VERSION_MAJOR) || defined(SDCC)
# define COMPILER_ID "SDCC"
# if defined(__SDCC_VERSION_MAJOR)
#  define COMPILER_VERSION_MAJOR DEC(__SDCC_VERSION_MAJOR)
#  define COMPILER_VERSION_MINOR DEC(__SDCC_VERSION_MINOR)
#  define COMPILER_VERSION_PATCH DEC(__SDCC_VERSION_PATCH)
# else
  /* SDCC = VRP */
#  define COMPILER_VERSION_MAJOR DEC(SDCC/100)
#  define COMPILER_VERSION_MINOR DEC(SDCC/10 % 10)
#  define COMPILER_VERSION_PATCH DEC(SDCC    % 10)
# endif


/* These compilers are either not known or too old to define an
  identification macro.  Try to identify the platform and guess that
  it is the native compiler.  */
#elif defined(__hpux) || defined(__hpua)
# define COMPILER_ID "HP"

#else /* unknown compiler */
# define COMPILER_ID ""
#endif

/* Construct the string literal in pieces to prevent the source from
   getting matched.  Store it in a pointer rather than an array
   because some compilers will just produce instructions to fill the
   array rather than assigning a pointer to a static array.  */
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]";
#ifdef SIMULATE_ID
char const* info_simulate = "INFO" ":" "simulate[" SIMULATE_ID "]";
#endif

#ifdef __QNXNTO__
char const* qnxnto = "INFO" ":" "qnxnto[]";
#endif

#if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
char const *info_cray = "INFO" ":" "compiler_wrapper[CrayPrgEnv]";
#endif

#define STRINGIFY_HELPER(X) #X
#define STRINGIFY(X) STRINGIFY_HELPER(X)

/* Identify known platforms by name.  */
#if defined(__linux) || defined(__linux__) || defined(linux)
# define PLATFORM_ID "Linux"

#elif defined(__CYGWIN__)
# define PLATFORM_ID "Cygwin"

#elif defined(__MINGW32__)
# define PLATFORM_ID "MinGW"

#elif defined(__APPLE__)
# define PLATFORM_ID "Darwin"

#elif defined(_WIN32) || defined(__WIN32__) || defined(WIN32)
```

```
# define PLATFORM_ID "Windows"

#elif defined(__FreeBSD__) || defined(__FreeBSD)
# define PLATFORM_ID "FreeBSD"

#elif defined(__NetBSD__) || defined(__NetBSD)
# define PLATFORM_ID "NetBSD"

#elif defined(__OpenBSD__) || defined(__OPENBSD)
# define PLATFORM_ID "OpenBSD"

#elif defined(__sun) || defined(sun)
# define PLATFORM_ID "SunOS"

#elif defined(_AIX) || defined(__AIX) || defined(__AIX__) || defined(__aix) || defined(__aix__)
# define PLATFORM_ID "AIX"

#elif defined(__hpux) || defined(__hpux__)
# define PLATFORM_ID "HP-UX"

#elif defined(__HAIKU__)
# define PLATFORM_ID "Haiku"

#elif defined(__BeOS) || defined(__BEOS__) || defined(_BEOS)
# define PLATFORM_ID "BeOS"

#elif defined(__QNX__) || defined(__QNXNTO__)
# define PLATFORM_ID "QNX"

#elif defined(__tru64) || defined(_tru64) || defined(__TRU64__)
# define PLATFORM_ID "Tru64"

#elif defined(__riscos) || defined(__riscos__)
# define PLATFORM_ID "RISCos"

#elif defined(__sinix) || defined(__sinix__) || defined(__SINIX__)
# define PLATFORM_ID "SINIX"

#elif defined(__UNIX_SV__)
# define PLATFORM_ID "UNIX_SV"

#elif defined(__bsdos__)
# define PLATFORM_ID "BSDOS"

#elif defined(_MPRAS) || defined(MPRAS)
# define PLATFORM_ID "MP-RAS"

#elif defined(__osf) || defined(__osf__)
# define PLATFORM_ID "OSF1"

#elif defined(_SCO_SV) || defined(SCO_SV) || defined(sco_sv)
# define PLATFORM_ID "SCO_SV"

#elif defined(__ultrix) || defined(__ultrix__) || defined(_ULTRIX)
# define PLATFORM_ID "ULTRIX"

#elif defined(__XENIX__) || defined(_XENIX) || defined(XENIX)
# define PLATFORM_ID "Xenix"

#elif defined(__WATCOMC__)
# if defined(__LINUX__)
#  define PLATFORM_ID "Linux"

# elif defined(__DOS__)
#  define PLATFORM_ID "DOS"

# elif defined(__OS2__)
#  define PLATFORM_ID "OS2"

# elif defined(__WINDOWS__)
#  define PLATFORM_ID "Windows3x"

# elif defined(__VXWORKS__)
#  define PLATFORM_ID "VxWorks"

# else /* unknown platform */
#  define PLATFORM_ID
# endif

#elif defined(__INTEGRITY)
# if defined(INT_178B)
#  define PLATFORM_ID "Integrity178"

# else /* regular Integrity */
#  define PLATFORM_ID "Integrity"
# endif

```

```
#else /* unknown platform */
# define PLATFORM_ID

#endif

/* For windows compilers MSVC and Intel we can determine
   the architecture of the compiler being used.  This is because
   the compilers do not have flags that can change the architecture,
   but rather depend on which compiler is being used
*/
#if defined(_WIN32) && defined(_MSC_VER)
# if defined(_M_IA64)
#  define ARCHITECTURE_ID "IA64"

# elif defined(_M_X64) || defined(_M_AMD64)
#  define ARCHITECTURE_ID "x64"

# elif defined(_M_IX86)
#  define ARCHITECTURE_ID "X86"

# elif defined(_M_ARM64)
#  define ARCHITECTURE_ID "ARM64"

# elif defined(_M_ARM)
#  if _M_ARM == 4
#   define ARCHITECTURE_ID "ARMV4I"
#  elif _M_ARM == 5
#   define ARCHITECTURE_ID "ARMV5I"
#  else
#   define ARCHITECTURE_ID "ARMV" STRINGIFY(_M_ARM)
#  endif

# elif defined(_M_MIPS)
#  define ARCHITECTURE_ID "MIPS"

# elif defined(_M_SH)
#  define ARCHITECTURE_ID "SHx"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#elif defined(__WATCOMC__)
# if defined(_M_I86)
#  define ARCHITECTURE_ID "I86"

# elif defined(_M_IX86)
#  define ARCHITECTURE_ID "X86"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#elif defined(__IAR_SYSTEMS_ICC__) || defined(__IAR_SYSTEMS_ICC)
# if defined(__ICCARM__)
#  define ARCHITECTURE_ID "ARM"

# elif defined(__ICCRX__)
#  define ARCHITECTURE_ID "RX"

# elif defined(__ICCRH850__)
#  define ARCHITECTURE_ID "RH850"

# elif defined(__ICCRL78__)
#  define ARCHITECTURE_ID "RL78"

# elif defined(__ICCRISCV__)
#  define ARCHITECTURE_ID "RISCV"

# elif defined(__ICCAVR__)
#  define ARCHITECTURE_ID "AVR"

# elif defined(__ICC430__)
#  define ARCHITECTURE_ID "MSP430"

# elif defined(__ICCV850__)
#  define ARCHITECTURE_ID "V850"

# elif defined(__ICC8051__)
#  define ARCHITECTURE_ID "8051"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#elif defined(__ghs__)
# if defined(__PPC64__)
#  define ARCHITECTURE_ID "PPC64"

# elif defined(__ppc__)
#  define ARCHITECTURE_ID "PPC"

# elif defined(__ARM__)
#  define ARCHITECTURE_ID "ARM"

# elif defined(__x86_64__)
#  define ARCHITECTURE_ID "x64"

# elif defined(__i386__)
#  define ARCHITECTURE_ID "X86"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#elif defined(__TI_COMPILER_VERSION__)
# if defined(__TI_ARM__)
#  define ARCHITECTURE_ID "ARM"

# elif defined(__MSP430__)
#  define ARCHITECTURE_ID "MSP430"

# elif defined(__TMS320C28XX__)
#  define ARCHITECTURE_ID "TMS320C28x"

# elif defined(__TMS320C6X__) || defined(_TMS320C6X)
#  define ARCHITECTURE_ID "TMS320C6x"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#else
#  define ARCHITECTURE_ID
#endif

/* Convert integer to decimal digit literals.  */
#define DEC(n)                   \
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))

/* Convert integer to hex digit literals.  */
#define HEX(n)             \
  ('0' + ((n)»28 & 0xF)), \
  ('0' + ((n)»24 & 0xF)), \
  ('0' + ((n)»20 & 0xF)), \
  ('0' + ((n)»16 & 0xF)), \
  ('0' + ((n)»12 & 0xF)), \
  ('0' + ((n)»8  & 0xF)), \
  ('0' + ((n)»4  & 0xF)), \
  ('0' + ((n)     & 0xF))

/* Construct a string literal encoding the version number components. */
#ifdef COMPILER_VERSION_MAJOR
char const info_version[] = {
  'I', 'N', 'F', 'O', ':',
  'c','o','m','p','i','l','e','r','_','v','e','r','s','i','o','n','[',
  COMPILER_VERSION_MAJOR,
# ifdef COMPILER_VERSION_MINOR
  '.', COMPILER_VERSION_MINOR,
#  ifdef COMPILER_VERSION_PATCH
   '.', COMPILER_VERSION_PATCH,
#   ifdef COMPILER_VERSION_TWEAK
    '.', COMPILER_VERSION_TWEAK,
#   endif
#  endif
# endif
  ']','\0'};
#endif

/* Construct a string literal encoding the internal version number. */
#ifdef COMPILER_VERSION_INTERNAL
char const info_version_internal[] = {
  'I', 'N', 'F', 'O', ':',
  'c','o','m','p','i','l','e','r','_','v','e','r','s','i','o','n','_',
  'i','n','t','e','r','n','a','l','[',
  COMPILER_VERSION_INTERNAL,']','\0'};
#endif
```

```

/* Construct a string literal encoding the version number components. */
#ifdef SIMULATE_VERSION_MAJOR
char const info_simulate_version[] = {
  'I', 'N', 'F', 'O', ':',
  's','i','m','u','l','a','t','e','_','v','e','r','s','i','o','n','[',
  SIMULATE_VERSION_MAJOR,
# ifdef SIMULATE_VERSION_MINOR
  '.', SIMULATE_VERSION_MINOR,
#  ifdef SIMULATE_VERSION_PATCH
   '.', SIMULATE_VERSION_PATCH,
#   ifdef SIMULATE_VERSION_TWEAK
    '.', SIMULATE_VERSION_TWEAK,
#   endif
#  endif
# endif
  ']','\0'};
#endif

/* Construct the string literal in pieces to prevent the source from
   getting matched.  Store it in a pointer rather than an array
   because some compilers will just produce instructions to fill the
   array rather than assigning a pointer to a static array.  */
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]";
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]";



#if !defined(__STDC__)
# if (defined(_MSC_VER) && !defined(__clang__)) \
  || (defined(__ibmxl__) || defined(__IBMC__))
#  define C_DIALECT "90"
# else
#  define C_DIALECT
# endif
#elif __STDC_VERSION__ >= 201000L
# define C_DIALECT "11"
#elif __STDC_VERSION__ >= 199901L
# define C_DIALECT "99"
#else
# define C_DIALECT "90"
#endif
const char* info_language_dialect_default =
  "INFO" ":" "dialect_default[" C_DIALECT "]";

/*--------------------------------------------------------------------------*/

#ifdef ID_VOID_MAIN
void main() {}
#else
# if defined(__CLASSIC_C__)
int main(argc, argv) int argc; char *argv[];
# else
int main(int argc, char* argv[])
# endif
{
  int require = 0;
  require += info_compiler[argc];
  require += info_platform[argc];
  require += info_arch[argc];
#ifdef COMPILER_VERSION_MAJOR
  require += info_version[argc];
#endif
#ifdef COMPILER_VERSION_INTERNAL
  require += info_version_internal[argc];
#endif
#ifdef SIMULATE_ID
  require += info_simulate[argc];
#endif
#ifdef SIMULATE_VERSION_MAJOR
  require += info_simulate_version[argc];
#endif
#if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
  require += info_cray[argc];
#endif
  require += info_language_dialect_default[argc];
  (void)argv;
  return require;
}
#endif
```

6.7 build/CMakeFiles/3.19.7/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

Macros

- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X) #X
- #define STRINGIFY(X) STRINGIFY_HELPER(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define CXX_STD __cplusplus

Functions

- int main (int argc, char ∗argv[ ])

Variables

- char const ∗ info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const ∗ info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const ∗ info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char ∗ info_language_dialect_default

6.7.1 Macro Definition Documentation

6.7.1.1 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

Definition at line 546 of file CMakeCXXCompilerId.cpp.

6.7.1.2 COMPILER_ID

```
#define COMPILER_ID ""
```

Definition at line 299 of file CMakeCXXCompilerId.cpp.

6.7.1.3 CXX_STD

```
#define CXX_STD __cplusplus
```

Definition at line 638 of file CMakeCXXCompilerId.cpp.

6.7.1.4 DEC

```
#define DEC(
              n )
```

Value:

```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```

Definition at line 550 of file CMakeCXXCompilerId.cpp.

6.7.1.5 HEX

```
#define HEX(
              n )
```

Value:

```
('0' + ((n)»28 & 0xF)), \
('0' + ((n)»24 & 0xF)), \
('0' + ((n)»20 & 0xF)), \
('0' + ((n)»16 & 0xF)), \
('0' + ((n)»12 & 0xF)), \
('0' + ((n)»8  & 0xF)), \
('0' + ((n)»4  & 0xF)), \
('0' + ((n)     & 0xF))
```

Definition at line 561 of file CMakeCXXCompilerId.cpp.

6.7.1.6 PLATFORM_ID

```
#define PLATFORM_ID
```

Definition at line 424 of file CMakeCXXCompilerId.cpp.

6.7.1.7 STRINGIFY

```
#define STRINGIFY(
              X ) STRINGIFY_HELPER(X)
```

Definition at line 320 of file CMakeCXXCompilerId.cpp.

6.7.1.8 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
              X ) #X
```

Definition at line 319 of file CMakeCXXCompilerId.cpp.

6.7.2 Function Documentation

6.7.2.1 main()

```
int main (
             int argc,
             char * argv[] )
```

Definition at line 657 of file CMakeCXXCompilerId.cpp.

6.7.3 Variable Documentation

6.7.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

Definition at line 621 of file CMakeCXXCompilerId.cpp.

6.7.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

Definition at line 306 of file CMakeCXXCompilerId.cpp.

6.7.3.3 info_language_dialect_default

```
const char* info_language_dialect_default
```

Initial value:

```
= "INFO" ":" "dialect_default["
  "98"
"]"
```

Definition at line 641 of file CMakeCXXCompilerId.cpp.

6.7.3.4 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

Definition at line 620 of file CMakeCXXCompilerId.cpp.

6.8 CMakeCXXCompilerId.cpp

```
/* This source file must have a .cpp extension so that all C++ compilers
   recognize the extension without flags.  Borland does not know .cxx for
   example.  */
#ifndef __cplusplus
# error "A C compiler has been selected for C++."
#endif


/* Version number components: V=Version, R=Revision, P=Patch
   Version date components:   YYYY=Year, MM=Month,   DD=Day  */

#if defined(__COMO__)
# define COMPILER_ID "Comeau"
  /* __COMO_VERSION__ = VRR */
# define COMPILER_VERSION_MAJOR DEC(__COMO_VERSION__ / 100)
# define COMPILER_VERSION_MINOR DEC(__COMO_VERSION__ % 100)

#elif defined(__INTEL_COMPILER) || defined(__ICC)
# define COMPILER_ID "Intel"
# if defined(_MSC_VER)
#  define SIMULATE_ID "MSVC"
# endif
# if defined(__GNUC__)
#  define SIMULATE_ID "GNU"
# endif
  /* __INTEL_COMPILER = VRP */
# define COMPILER_VERSION_MAJOR DEC(__INTEL_COMPILER/100)
# define COMPILER_VERSION_MINOR DEC(__INTEL_COMPILER/10 % 10)
# if defined(__INTEL_COMPILER_UPDATE)
#  define COMPILER_VERSION_PATCH DEC(__INTEL_COMPILER_UPDATE)
# else
#  define COMPILER_VERSION_PATCH DEC(__INTEL_COMPILER   % 10)
# endif
# if defined(__INTEL_COMPILER_BUILD_DATE)
  /* __INTEL_COMPILER_BUILD_DATE = YYYYMMDD */
#  define COMPILER_VERSION_TWEAK DEC(__INTEL_COMPILER_BUILD_DATE)
# endif
# if defined(_MSC_VER)
   /* _MSC_VER = VVRR */
#  define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
#  define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
# endif
# if defined(__GNUC__)
#  define SIMULATE_VERSION_MAJOR DEC(__GNUC__)
# elif defined(__GNUG__)
#  define SIMULATE_VERSION_MAJOR DEC(__GNUG__)
# endif
# if defined(__GNUC_MINOR__)
#  define SIMULATE_VERSION_MINOR DEC(__GNUC_MINOR__)
# endif
# if defined(__GNUC_PATCHLEVEL__)
#  define SIMULATE_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
# endif

#elif defined(__PATHCC__)
# define COMPILER_ID "PathScale"
# define COMPILER_VERSION_MAJOR DEC(__PATHCC__)
# define COMPILER_VERSION_MINOR DEC(__PATHCC_MINOR__)
# if defined(__PATHCC_PATCHLEVEL__)
#  define COMPILER_VERSION_PATCH DEC(__PATHCC_PATCHLEVEL__)
# endif

#elif defined(__BORLANDC__) && defined(__CODEGEARC_VERSION__)
```

```
# define COMPILER_ID "Embarcadero"
# define COMPILER_VERSION_MAJOR HEX(__CODEGEARC_VERSION__»24 & 0x00FF)
# define COMPILER_VERSION_MINOR HEX(__CODEGEARC_VERSION__»16 & 0x00FF)
# define COMPILER_VERSION_PATCH DEC(__CODEGEARC_VERSION__     & 0xFFFF)

#elif defined(__BORLANDC__)
# define COMPILER_ID "Borland"
  /* __BORLANDC__ = 0xVRR */
# define COMPILER_VERSION_MAJOR HEX(__BORLANDC__»8)
# define COMPILER_VERSION_MINOR HEX(__BORLANDC__ & 0xFF)

#elif defined(__WATCOMC__) && __WATCOMC__ < 1200
# define COMPILER_ID "Watcom"
   /* __WATCOMC__ = VVRR */
# define COMPILER_VERSION_MAJOR DEC(__WATCOMC__ / 100)
# define COMPILER_VERSION_MINOR DEC((__WATCOMC__ / 10) % 10)
# if (__WATCOMC__ % 10) > 0
#  define COMPILER_VERSION_PATCH DEC(__WATCOMC__ % 10)
# endif

#elif defined(__WATCOMC__)
# define COMPILER_ID "OpenWatcom"
   /* __WATCOMC__ = VVRP + 1100 */
# define COMPILER_VERSION_MAJOR DEC((__WATCOMC__ - 1100) / 100)
# define COMPILER_VERSION_MINOR DEC((__WATCOMC__ / 10) % 10)
# if (__WATCOMC__ % 10) > 0
#  define COMPILER_VERSION_PATCH DEC(__WATCOMC__ % 10)
# endif

#elif defined(__SUNPRO_CC)
# define COMPILER_ID "SunPro"
# if __SUNPRO_CC >= 0x5100
   /* __SUNPRO_CC = 0xVRRP */
#  define COMPILER_VERSION_MAJOR HEX(__SUNPRO_CC»12)
#  define COMPILER_VERSION_MINOR HEX(__SUNPRO_CC»4 & 0xFF)
#  define COMPILER_VERSION_PATCH HEX(__SUNPRO_CC    & 0xF)
# else
   /* __SUNPRO_CC = 0xVRP */
#  define COMPILER_VERSION_MAJOR HEX(__SUNPRO_CC»8)
#  define COMPILER_VERSION_MINOR HEX(__SUNPRO_CC»4 & 0xF)
#  define COMPILER_VERSION_PATCH HEX(__SUNPRO_CC    & 0xF)
# endif

#elif defined(__HP_aCC)
# define COMPILER_ID "HP"
  /* __HP_aCC = VVRRPP */
# define COMPILER_VERSION_MAJOR DEC(__HP_aCC/10000)
# define COMPILER_VERSION_MINOR DEC(__HP_aCC/100 % 100)
# define COMPILER_VERSION_PATCH DEC(__HP_aCC     % 100)

#elif defined(__DECCXX)
# define COMPILER_ID "Compaq"
  /* __DECCXX_VER = VVRRTPPPP */
# define COMPILER_VERSION_MAJOR DEC(__DECCXX_VER/10000000)
# define COMPILER_VERSION_MINOR DEC(__DECCXX_VER/100000  % 100)
# define COMPILER_VERSION_PATCH DEC(__DECCXX_VER         % 10000)

#elif defined(__IBMCPP__) && defined(__COMPILER_VER__)
# define COMPILER_ID "zOS"
  /* __IBMCPP__ = VRP */
# define COMPILER_VERSION_MAJOR DEC(__IBMCPP__/100)
# define COMPILER_VERSION_MINOR DEC(__IBMCPP__/10 % 10)
# define COMPILER_VERSION_PATCH DEC(__IBMCPP__    % 10)

#elif defined(__ibmxl__) && defined(__clang__)
# define COMPILER_ID "XLClang"
# define COMPILER_VERSION_MAJOR DEC(__ibmxl_version__)
# define COMPILER_VERSION_MINOR DEC(__ibmxl_release__)
# define COMPILER_VERSION_PATCH DEC(__ibmxl_modification__)
# define COMPILER_VERSION_TWEAK DEC(__ibmxl_ptf_fix_level__)


#elif defined(__IBMCPP__) && !defined(__COMPILER_VER__) && __IBMCPP__ >= 800
# define COMPILER_ID "XL"
  /* __IBMCPP__ = VRP */
# define COMPILER_VERSION_MAJOR DEC(__IBMCPP__/100)
# define COMPILER_VERSION_MINOR DEC(__IBMCPP__/10 % 10)
# define COMPILER_VERSION_PATCH DEC(__IBMCPP__    % 10)

#elif defined(__IBMCPP__) && !defined(__COMPILER_VER__) && __IBMCPP__ < 800
# define COMPILER_ID "VisualAge"
  /* __IBMCPP__ = VRP */
# define COMPILER_VERSION_MAJOR DEC(__IBMCPP__/100)
# define COMPILER_VERSION_MINOR DEC(__IBMCPP__/10 % 10)
# define COMPILER_VERSION_PATCH DEC(__IBMCPP__    % 10)

#elif defined(__PGI)
```

```
# define COMPILER_ID "PGI"
# define COMPILER_VERSION_MAJOR DEC(__PGIC__)
# define COMPILER_VERSION_MINOR DEC(__PGIC_MINOR__)
# if defined(__PGIC_PATCHLEVEL__)
#  define COMPILER_VERSION_PATCH DEC(__PGIC_PATCHLEVEL__)
# endif

#elif defined(_CRAYC)
# define COMPILER_ID "Cray"
# define COMPILER_VERSION_MAJOR DEC(_RELEASE_MAJOR)
# define COMPILER_VERSION_MINOR DEC(_RELEASE_MINOR)

#elif defined(__TI_COMPILER_VERSION__)
# define COMPILER_ID "TI"
  /* __TI_COMPILER_VERSION__ = VVVRRRPPP */
# define COMPILER_VERSION_MAJOR DEC(__TI_COMPILER_VERSION__/1000000)
# define COMPILER_VERSION_MINOR DEC(__TI_COMPILER_VERSION__/1000   % 1000)
# define COMPILER_VERSION_PATCH DEC(__TI_COMPILER_VERSION__        % 1000)

#elif defined(__FUJITSU) || defined(__FCC_VERSION) || defined(__fcc_version)
# define COMPILER_ID "Fujitsu"

#elif defined(__ghs__)
# define COMPILER_ID "GHS"
/* __GHS_VERSION_NUMBER = VVVVRP */
# ifdef __GHS_VERSION_NUMBER
# define COMPILER_VERSION_MAJOR DEC(__GHS_VERSION_NUMBER / 100)
# define COMPILER_VERSION_MINOR DEC(__GHS_VERSION_NUMBER / 10 % 10)
# define COMPILER_VERSION_PATCH DEC(__GHS_VERSION_NUMBER      % 10)
# endif

#elif defined(__SCO_VERSION__)
# define COMPILER_ID "SCO"

#elif defined(__ARMCC_VERSION) && !defined(__clang__)
# define COMPILER_ID "ARMCC"
#if __ARMCC_VERSION >= 1000000
  /* __ARMCC_VERSION = VRRPPPP */
  # define COMPILER_VERSION_MAJOR DEC(__ARMCC_VERSION/1000000)
  # define COMPILER_VERSION_MINOR DEC(__ARMCC_VERSION/10000 % 100)
  # define COMPILER_VERSION_PATCH DEC(__ARMCC_VERSION     % 10000)
#else
  /* __ARMCC_VERSION = VRPPPP */
  # define COMPILER_VERSION_MAJOR DEC(__ARMCC_VERSION/100000)
  # define COMPILER_VERSION_MINOR DEC(__ARMCC_VERSION/10000 % 10)
  # define COMPILER_VERSION_PATCH DEC(__ARMCC_VERSION    % 10000)
#endif


#elif defined(__clang__) && defined(__apple_build_version__)
# define COMPILER_ID "AppleClang"
# if defined(_MSC_VER)
#  define SIMULATE_ID "MSVC"
# endif
# define COMPILER_VERSION_MAJOR DEC(__clang_major__)
# define COMPILER_VERSION_MINOR DEC(__clang_minor__)
# define COMPILER_VERSION_PATCH DEC(__clang_patchlevel__)
# if defined(_MSC_VER)
   /* _MSC_VER = VVRR */
#  define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
#  define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
# endif
# define COMPILER_VERSION_TWEAK DEC(__apple_build_version__)

#elif defined(__clang__) && defined(__ARMCOMPILER_VERSION)
# define COMPILER_ID "ARMClang"
  # define COMPILER_VERSION_MAJOR DEC(__ARMCOMPILER_VERSION/1000000)
  # define COMPILER_VERSION_MINOR DEC(__ARMCOMPILER_VERSION/10000 % 100)
  # define COMPILER_VERSION_PATCH DEC(__ARMCOMPILER_VERSION     % 10000)
# define COMPILER_VERSION_INTERNAL DEC(__ARMCOMPILER_VERSION)

#elif defined(__clang__)
# define COMPILER_ID "Clang"
# if defined(_MSC_VER)
#  define SIMULATE_ID "MSVC"
# endif
# define COMPILER_VERSION_MAJOR DEC(__clang_major__)
# define COMPILER_VERSION_MINOR DEC(__clang_minor__)
# define COMPILER_VERSION_PATCH DEC(__clang_patchlevel__)
# if defined(_MSC_VER)
   /* _MSC_VER = VVRR */
#  define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
#  define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
# endif

#elif defined(__GNUC__) || defined(__GNUG__)
# define COMPILER_ID "GNU"
# if defined(__GNUC__)
#  define COMPILER_VERSION_MAJOR DEC(__GNUC__)
# else
#  define COMPILER_VERSION_MAJOR DEC(__GNUG__)
# endif
# if defined(__GNUC_MINOR__)
#  define COMPILER_VERSION_MINOR DEC(__GNUC_MINOR__)
# endif
# if defined(__GNUC_PATCHLEVEL__)
#  define COMPILER_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
# endif

#elif defined(_MSC_VER)
# define COMPILER_ID "MSVC"
  /* _MSC_VER = VVRR */
# define COMPILER_VERSION_MAJOR DEC(_MSC_VER / 100)
# define COMPILER_VERSION_MINOR DEC(_MSC_VER % 100)
# if defined(_MSC_FULL_VER)
#  if _MSC_VER >= 1400
    /* _MSC_FULL_VER = VVRRPPPPP */
#   define COMPILER_VERSION_PATCH DEC(_MSC_FULL_VER % 100000)
#  else
    /* _MSC_FULL_VER = VVRRPPPP */
#   define COMPILER_VERSION_PATCH DEC(_MSC_FULL_VER % 10000)
#  endif
# endif
# if defined(_MSC_BUILD)
#  define COMPILER_VERSION_TWEAK DEC(_MSC_BUILD)
# endif

#elif defined(__VISUALDSPVERSION__) || defined(__ADSPBLACKFIN__) || defined(__ADSPTS__) ||
 defined(__ADSP21000__)
# define COMPILER_ID "ADSP"
#if defined(__VISUALDSPVERSION__)
  /* __VISUALDSPVERSION__ = 0xVVRRPP00 */
# define COMPILER_VERSION_MAJOR HEX(__VISUALDSPVERSION__»24)
# define COMPILER_VERSION_MINOR HEX(__VISUALDSPVERSION__»16 & 0xFF)
# define COMPILER_VERSION_PATCH HEX(__VISUALDSPVERSION__»8  & 0xFF)
#endif

#elif defined(__IAR_SYSTEMS_ICC__) || defined(__IAR_SYSTEMS_ICC)
# define COMPILER_ID "IAR"
# if defined(__VER__) && defined(__ICCARM__)
#  define COMPILER_VERSION_MAJOR DEC((__VER__) / 1000000)
#  define COMPILER_VERSION_MINOR DEC(((__VER__) / 1000) % 1000)
#  define COMPILER_VERSION_PATCH DEC((__VER__) % 1000)
#  define COMPILER_VERSION_INTERNAL DEC(__IAR_SYSTEMS_ICC__)
# elif defined(__VER__) && (defined(__ICCAVR__) || defined(__ICCRX__) || defined(__ICCRH850__) ||
 defined(__ICCRL78__) || defined(__ICC430__) || defined(__ICCRISCV__) || defined(__ICCV850__) ||
 defined(__ICC8051__))
#  define COMPILER_VERSION_MAJOR DEC((__VER__) / 100)
#  define COMPILER_VERSION_MINOR DEC((__VER__) - (((__VER__) / 100)*100))
#  define COMPILER_VERSION_PATCH DEC(__SUBVERSION__)
#  define COMPILER_VERSION_INTERNAL DEC(__IAR_SYSTEMS_ICC__)
# endif


/* These compilers are either not known or too old to define an
  identification macro.  Try to identify the platform and guess that
  it is the native compiler.  */
#elif defined(__hpux) || defined(__hpua)
# define COMPILER_ID "HP"

#else /* unknown compiler */
# define COMPILER_ID ""
#endif

/* Construct the string literal in pieces to prevent the source from
   getting matched.  Store it in a pointer rather than an array
   because some compilers will just produce instructions to fill the
   array rather than assigning a pointer to a static array.  */
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]";
#ifdef SIMULATE_ID
char const* info_simulate = "INFO" ":" "simulate[" SIMULATE_ID "]";
#endif

#ifdef __QNXNTO__
char const* qnxnto = "INFO" ":" "qnxnto[]";
#endif

#if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
char const *info_cray = "INFO" ":" "compiler_wrapper[CrayPrgEnv]";
#endif

#define STRINGIFY_HELPER(X) #X
#define STRINGIFY(X) STRINGIFY_HELPER(X)

```

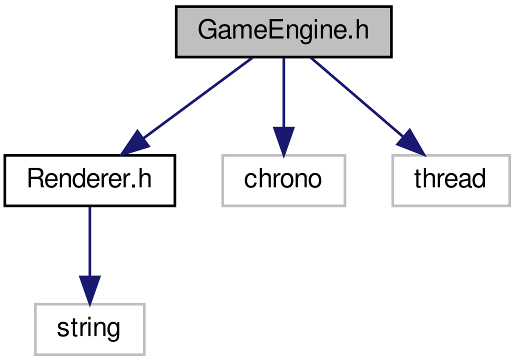
```
/* Identify known platforms by name.  */
#if defined(__linux) || defined(__linux__) || defined(linux)
# define PLATFORM_ID "Linux"

#elif defined(__CYGWIN__)
# define PLATFORM_ID "Cygwin"

#elif defined(__MINGW32__)
# define PLATFORM_ID "MinGW"

#elif defined(__APPLE__)
# define PLATFORM_ID "Darwin"

#elif defined(_WIN32) || defined(__WIN32__) || defined(WIN32)
# define PLATFORM_ID "Windows"

#elif defined(__FreeBSD__) || defined(__FreeBSD)
# define PLATFORM_ID "FreeBSD"

#elif defined(__NetBSD__) || defined(__NetBSD)
# define PLATFORM_ID "NetBSD"

#elif defined(__OpenBSD__) || defined(__OPENBSD)
# define PLATFORM_ID "OpenBSD"

#elif defined(__sun) || defined(sun)
# define PLATFORM_ID "SunOS"

#elif defined(_AIX) || defined(__AIX) || defined(__AIX__) || defined(__aix) || defined(__aix__)
# define PLATFORM_ID "AIX"

#elif defined(__hpux) || defined(__hpux__)
# define PLATFORM_ID "HP-UX"

#elif defined(__HAIKU__)
# define PLATFORM_ID "Haiku"

#elif defined(__BeOS) || defined(__BEOS__) || defined(_BEOS)
# define PLATFORM_ID "BeOS"

#elif defined(__QNX__) || defined(__QNXNTO__)
# define PLATFORM_ID "QNX"

#elif defined(__tru64) || defined(_tru64) || defined(__TRU64__)
# define PLATFORM_ID "Tru64"

#elif defined(__riscos) || defined(__riscos__)
# define PLATFORM_ID "RISCos"

#elif defined(__sinix) || defined(__sinix__) || defined(__SINIX__)
# define PLATFORM_ID "SINIX"

#elif defined(__UNIX_SV__)
# define PLATFORM_ID "UNIX_SV"

#elif defined(__bsdos__)
# define PLATFORM_ID "BSDOS"

#elif defined(_MPRAS) || defined(MPRAS)
# define PLATFORM_ID "MP-RAS"

#elif defined(__osf) || defined(__osf__)
# define PLATFORM_ID "OSF1"

#elif defined(_SCO_SV) || defined(SCO_SV) || defined(sco_sv)
# define PLATFORM_ID "SCO_SV"

#elif defined(__ultrix) || defined(__ultrix__) || defined(_ULTRIX)
# define PLATFORM_ID "ULTRIX"

#elif defined(__XENIX__) || defined(_XENIX) || defined(XENIX)
# define PLATFORM_ID "Xenix"

#elif defined(__WATCOMC__)
# if defined(__LINUX__)
#  define PLATFORM_ID "Linux"

# elif defined(__DOS__)
#  define PLATFORM_ID "DOS"

# elif defined(__OS2__)
#  define PLATFORM_ID "OS2"

# elif defined(__WINDOWS__)
#  define PLATFORM_ID "Windows3x"

# elif defined(__VXWORKS__)
#  define PLATFORM_ID "VxWorks"

# else /* unknown platform */
#  define PLATFORM_ID
# endif

#elif defined(__INTEGRITY)
# if defined(INT_178B)
#  define PLATFORM_ID "Integrity178"

# else /* regular Integrity */
#  define PLATFORM_ID "Integrity"
# endif

#else /* unknown platform */
# define PLATFORM_ID

#endif

/* For windows compilers MSVC and Intel we can determine
   the architecture of the compiler being used.  This is because
   the compilers do not have flags that can change the architecture,
   but rather depend on which compiler is being used
*/
#if defined(_WIN32) && defined(_MSC_VER)
# if defined(_M_IA64)
#  define ARCHITECTURE_ID "IA64"

# elif defined(_M_X64) || defined(_M_AMD64)
#  define ARCHITECTURE_ID "x64"

# elif defined(_M_IX86)
#  define ARCHITECTURE_ID "X86"

# elif defined(_M_ARM64)
#  define ARCHITECTURE_ID "ARM64"

# elif defined(_M_ARM)
#  if _M_ARM == 4
#   define ARCHITECTURE_ID "ARMV4I"
#  elif _M_ARM == 5
#   define ARCHITECTURE_ID "ARMV5I"
#  else
#   define ARCHITECTURE_ID "ARMV" STRINGIFY(_M_ARM)
#  endif

# elif defined(_M_MIPS)
#  define ARCHITECTURE_ID "MIPS"

# elif defined(_M_SH)
#  define ARCHITECTURE_ID "SHx"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#elif defined(__WATCOMC__)
# if defined(_M_I86)
#  define ARCHITECTURE_ID "I86"

# elif defined(_M_IX86)
#  define ARCHITECTURE_ID "X86"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#elif defined(__IAR_SYSTEMS_ICC__) || defined(__IAR_SYSTEMS_ICC)
# if defined(__ICCARM__)
#  define ARCHITECTURE_ID "ARM"

# elif defined(__ICCRX__)
#  define ARCHITECTURE_ID "RX"

# elif defined(__ICCRH850__)
#  define ARCHITECTURE_ID "RH850"

# elif defined(__ICCRL78__)
#  define ARCHITECTURE_ID "RL78"

# elif defined(__ICCRISCV__)
#  define ARCHITECTURE_ID "RISCV"

# elif defined(__ICCAVR__)
#  define ARCHITECTURE_ID "AVR"

# elif defined(__ICC430__)
#  define ARCHITECTURE_ID "MSP430"

# elif defined(__ICCV850__)
#  define ARCHITECTURE_ID "V850"

# elif defined(__ICC8051__)
#  define ARCHITECTURE_ID "8051"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#elif defined(__ghs__)
# if defined(__PPC64__)
#  define ARCHITECTURE_ID "PPC64"

# elif defined(__ppc__)
#  define ARCHITECTURE_ID "PPC"

# elif defined(__ARM__)
#  define ARCHITECTURE_ID "ARM"

# elif defined(__x86_64__)
#  define ARCHITECTURE_ID "x64"

# elif defined(__i386__)
#  define ARCHITECTURE_ID "X86"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#elif defined(__TI_COMPILER_VERSION__)
# if defined(__TI_ARM__)
#  define ARCHITECTURE_ID "ARM"

# elif defined(__MSP430__)
#  define ARCHITECTURE_ID "MSP430"

# elif defined(__TMS320C28XX__)
#  define ARCHITECTURE_ID "TMS320C28x"

# elif defined(__TMS320C6X__) || defined(_TMS320C6X)
#  define ARCHITECTURE_ID "TMS320C6x"

# else /* unknown architecture */
#  define ARCHITECTURE_ID ""
# endif

#else
#  define ARCHITECTURE_ID
#endif

/* Convert integer to decimal digit literals.  */
#define DEC(n)                   \
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))

/* Convert integer to hex digit literals.  */
#define HEX(n)             \
  ('0' + ((n)»28 & 0xF)), \
  ('0' + ((n)»24 & 0xF)), \
  ('0' + ((n)»20 & 0xF)), \
  ('0' + ((n)»16 & 0xF)), \
  ('0' + ((n)»12 & 0xF)), \
  ('0' + ((n)»8  & 0xF)), \
  ('0' + ((n)»4  & 0xF)), \
  ('0' + ((n)     & 0xF))

/* Construct a string literal encoding the version number components. */
#ifdef COMPILER_VERSION_MAJOR
char const info_version[] = {
  'I', 'N', 'F', 'O', ':',
  'c','o','m','p','i','l','e','r','_','v','e','r','s','i','o','n','[',
  COMPILER_VERSION_MAJOR,
# ifdef COMPILER_VERSION_MINOR
  '.', COMPILER_VERSION_MINOR,
#  ifdef COMPILER_VERSION_PATCH
   '.', COMPILER_VERSION_PATCH,
#   ifdef COMPILER_VERSION_TWEAK
    '.', COMPILER_VERSION_TWEAK,
#   endif
#  endif
# endif
  ']','\0'};
#endif

/* Construct a string literal encoding the internal version number. */
#ifdef COMPILER_VERSION_INTERNAL
char const info_version_internal[] = {
  'I', 'N', 'F', 'O', ':',
  'c','o','m','p','i','l','e','r','_','v','e','r','s','i','o','n','_',
  'i','n','t','e','r','n','a','l','[',
  COMPILER_VERSION_INTERNAL,']','\0'};
#endif

/* Construct a string literal encoding the version number components. */
#ifdef SIMULATE_VERSION_MAJOR
char const info_simulate_version[] = {
  'I', 'N', 'F', 'O', ':',
  's','i','m','u','l','a','t','e','_','v','e','r','s','i','o','n','[',
  SIMULATE_VERSION_MAJOR,
# ifdef SIMULATE_VERSION_MINOR
  '.', SIMULATE_VERSION_MINOR,
#  ifdef SIMULATE_VERSION_PATCH
   '.', SIMULATE_VERSION_PATCH,
#   ifdef SIMULATE_VERSION_TWEAK
    '.', SIMULATE_VERSION_TWEAK,
#   endif
#  endif
# endif
  ']','\0'};
#endif

/* Construct the string literal in pieces to prevent the source from
   getting matched.  Store it in a pointer rather than an array
   because some compilers will just produce instructions to fill the
   array rather than assigning a pointer to a static array.  */
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]";
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]";



#if defined(__INTEL_COMPILER) && defined(_MSVC_LANG) && _MSVC_LANG < 201403L
#  if defined(__INTEL_CXX11_MODE__)
#    if defined(__cpp_aggregate_nsdmi)
#      define CXX_STD 201402L
#    else
#      define CXX_STD 201103L
#    endif
#  else
#    define CXX_STD 199711L
#  endif
#elif defined(_MSC_VER) && defined(_MSVC_LANG)
#  define CXX_STD _MSVC_LANG
#else
#  define CXX_STD __cplusplus
#endif

const char* info_language_dialect_default = "INFO" ":" "dialect_default["
#if CXX_STD > 201703L
  "20"
#elif CXX_STD >= 201703L
  "17"
#elif CXX_STD >= 201402L
  "14"
#elif CXX_STD >= 201103L
  "11"
#else
  "98"
#endif
"]";

/*--------------------------------------------------------------------------*/

int main(int argc, char* argv[])
{
  int require = 0;
  require += info_compiler[argc];
  require += info_platform[argc];
#ifdef COMPILER_VERSION_MAJOR
  require += info_version[argc];
#endif
#ifdef COMPILER_VERSION_INTERNAL
  require += info_version_internal[argc];
#endif
#ifdef SIMULATE_ID
  require += info_simulate[argc];
```

```
#endif
#ifdef SIMULATE_VERSION_MAJOR
  require += info_simulate_version[argc];
#endif
#if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
  require += info_cray[argc];
#endif
  require += info_language_dialect_default[argc];
  (void)argv;
  return require;
}
```

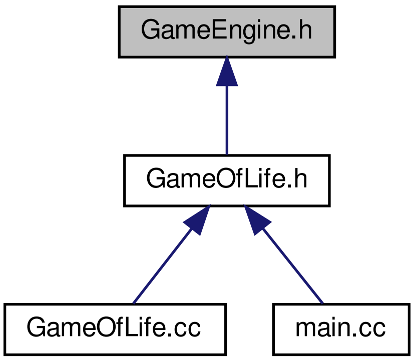
6.9 GameEngine.h File Reference

```
#include "Renderer.h"
#include <chrono>
#include <thread>
```

Include dependency graph for GameEngine.h:



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



Classes

- class GameEngine

  Base class for custom game engines.
- struct GameEngine::Config

  Config for game engines.

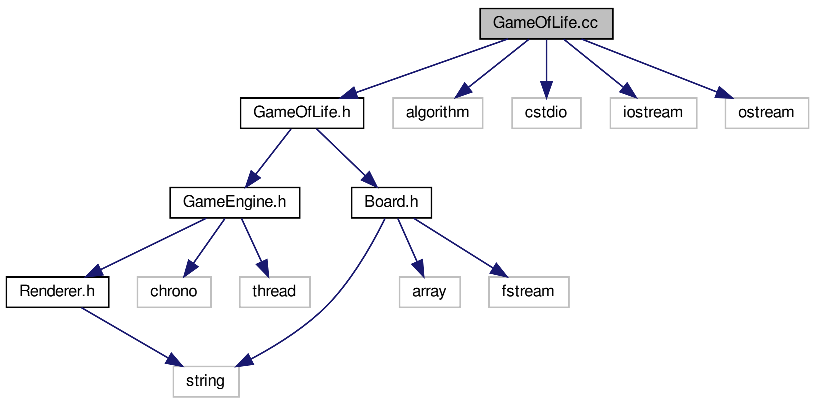
6.10 GameEngine.h

```
//
// Created by mateu on 3/22/2021.
//

#ifndef GAME_OF_LIFE_GAMEENGINE_H
#define GAME_OF_LIFE_GAMEENGINE_H

#include "Renderer.h"
#include <chrono>
#include <thread>

class GameEngine {

public:
    GameEngine() = delete;

    GameEngine(const GameEngine &) = delete;

    GameEngine &operator=(const GameEngine &) = delete;

    struct Config {
        int framerate;
        Renderer *renderer;
    } current_config_;

protected:

    explicit GameEngine(const GameEngine::Config &config) :
            current_config_(config),
            renderer_(config.renderer),
            running_(false) {};

    virtual void start_engine() final {
        on_start();
        while (running_) {
            // we are measuring the time before the work in the frame
            auto start = std::chrono::high_resolution_clock::now();
            on_tick();
            // we are measuring the time after the work in the frame
            auto stop = std::chrono::high_resolution_clock::now();
            // how long should the frame take
            auto target_frame_time = std::chrono::seconds(1 / current_config_.framerate);
            // how long it took
            auto current_frame_time = start - stop;
            // sleep for the difference between target and real time
            std::this_thread::sleep_for(target_frame_time - current_frame_time);
        }
        on_end();
    };

    virtual void on_start() = 0;

    virtual void on_tick() = 0;

    virtual void on_end() = 0;

    virtual ~GameEngine() { delete renderer_; }

protected:
    void start_game_loop() { running_ = true; }

    void stop_game_loop() { running_ = false; }

    Renderer *renderer_;

private:
    bool running_;

};


#endif //GAME_OF_LIFE_GAMEENGINE_H
```

6.11 GameOfLife.cc File Reference

```
#include "GameOfLife.h"
#include <algorithm>
#include <cstdio>
#include <iostream>
#include <ostream>
```

Include dependency graph for GameOfLife.cc:



Functions

- bool conway_activation (bool is_alive, int no_neighbours)

6.11.1 Function Documentation

6.11.1.1 conway_activation()

```
bool conway_activation (
            bool is_alive,
            int no_neighbours )
```

Activation function proposed by Conway in his original game

Parameters

is_alive	is cell that is checked alive
no_neighbours	how many neighbours are alive

Returns

should the cell be alive or not

Definition at line 112 of file GameOfLife.cc.

6.12 GameOfLife.cc

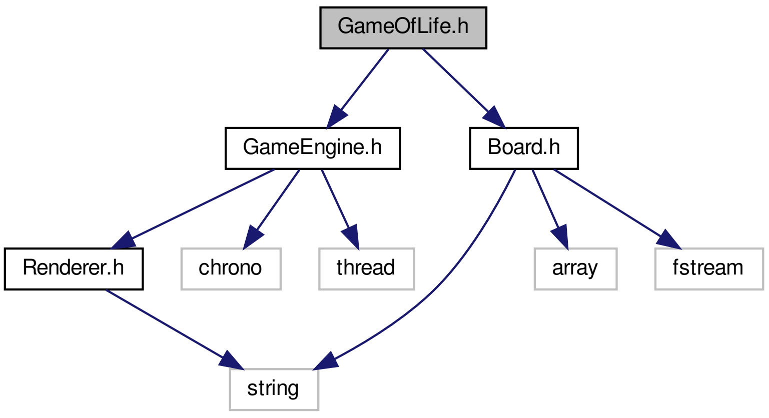
```
00001 //
00002 // Created by mateu on 3/24/2021.
00003 //
00004
```

```
#include "GameOfLife.h"
#include <algorithm>
#include <cstdio>
#include <iostream>
#include <ostream>
#include <cstdio>

static void show_board(const Board &board) {
  for (int i = 0; i < board.size(); i++) {
    std::cout « board(i) « " ";
  }
}

void GameOfLife::play() {
    start_engine();
}

void GameOfLife::set_activation_function(bool (*func)(bool, int)) {
    activation_func_ = func;
}

GameOfLife::~GameOfLife() {
    delete current_board_;
    delete next_board_;
    current_board_ = nullptr;
    next_board_ = nullptr;
}

void GameOfLife::on_start() {
    // Show a welcome screen and ready to go
    renderer_->create_window(current_board_->size_x(),
                             current_board_->size_y());

//    renderer_->clear_screen(Color::Black);
//    renderer_->show_text_big(Coord{0, 0}, "Gra");
//    renderer_->show_text_big(Coord{1, 0}, "w");
//    renderer_->show_text_big(Coord{2, 0}, "zycie");
//    renderer_->render();
//    std::string text;
//    while (text != "start") {
//        std::cin » text;
//    }
    start_game_loop();
}

void GameOfLife::render_current_board() {
    for (int x = 0; x < current_board_->size_x(); x++) {
        for (int y = 0; y < current_board_->size_y(); y++) {
            Color color;
            if ((*current_board_)(x, y))
                color = Color::White;
            else
                color = Color::Black;

            renderer_->set_pixel(Coord(x, y), color);
        }
    }
    renderer_->render();
}

void GameOfLife::on_tick() {
  // loop through all the active and inactive cells
  for (int i = 0; i < current_board_->size(); ++i) {
    // get neighbours for current cell
    auto neighbours = current_board_->get_neighbours(i);
    // make a cell alive if activation function determines so
    bool value = activation_func_(
        // Whats the current state of the cell
        (*next_board_)(i),
        // Count cells that are active around this cell
        std::count(neighbours.begin(), neighbours.end(), true));

#if DEBUG
    std::printf("Value at cell %d is %d input was (%lld)\n", i, value,
                std::count(neighbours.begin(), neighbours.end(), true));
#endif

    (*next_board_)(i) = value;
  }

#if DEBUG
  static int tick = 0;
  std::printf("Tick %d\n", tick++);
  std::printf("Current board\n");
  show_board(*current_board_);
  std::printf("\nNext board\n");
  show_board(*next_board_);
  std::printf("\n\n");
#else
  render_current_board();
#endif

  Board temp = *next_board_;
  *next_board_ = *current_board_;
  *current_board_ = temp;
}

void GameOfLife::on_end() {
    // Show some stats on exit
    renderer_->show_text_small(Coord(0, 0), "Dziekuje!");
}

bool conway_activation(bool is_alive, int no_neighbours) {
    if (is_alive) {
        if (no_neighbours == 2 || no_neighbours == 3)
            return true;
        else
            return false;
    } else {
        if (no_neighbours == 3)
            return true;
        else
            return false;
    }
}
```

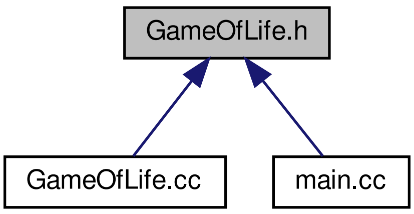
6.13 GameOfLife.h File Reference

```
#include "GameEngine.h"
#include "Board.h"
```

Include dependency graph for GameOfLife.h:



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



Classes

- class GameOfLife

  Implementation of the game of life.

Functions

- bool conway_activation (bool is_alive, int no_neighbours)

6.13.1 Function Documentation

6.13.1.1 conway_activation()

```
bool conway_activation (
            bool is_alive,
            int no_neighbours )
```

Activation function based on the original Conway's Game of Life

Parameters

is_alive	is the cell alive or not
no_neighbours	how many alive neighbours are around

Returns

true if cell should be alive false if not

Activation function proposed by Conway in his original game

Parameters

is_alive	is cell that is checked alive
no_neighbours	how many neighbours are alive

Returns

should the cell be alive or not

Definition at line 112 of file GameOfLife.cc.

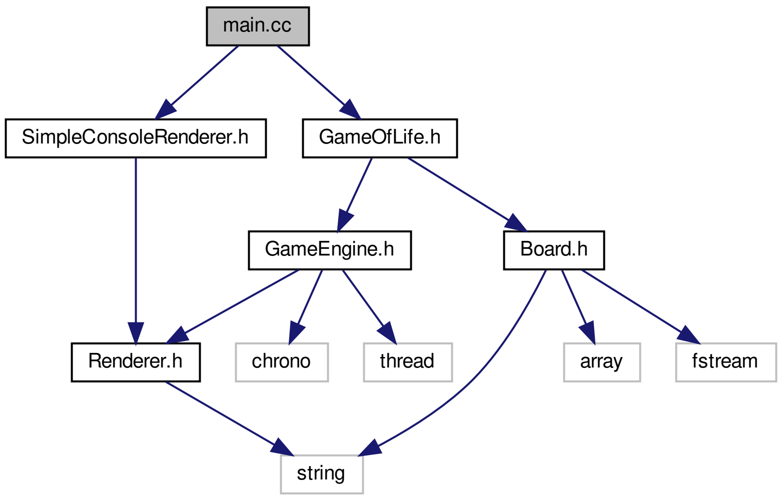
6.14 GameOfLife.h

```
//
// Created by mateu on 3/24/2021.
//

#ifndef GAME_OF_LIFE_GAMEOFLIFE_H
#define GAME_OF_LIFE_GAMEOFLIFE_H

#include "GameEngine.h"
#include "Board.h"

bool conway_activation(bool is_alive, int no_neighbours);

class GameOfLife : public GameEngine {
public:
    explicit GameOfLife(const Board &board, const Config &config)
            : GameEngine(config),
              activation_func_(conway_activation) {

        current_board_ = new Board(board);
        next_board_ = new Board(board);
    };


    GameOfLife(const GameOfLife &) = delete;

    const GameOfLife &operator=(const GameOfLife &) = delete;

    void render_current_board();

    void play();

    void set_activation_function(bool (*func)(bool, int));

    ~GameOfLife() override;

protected:


    void on_start() override;

    void on_tick() override;

    void on_end() override;

private:

    Board *current_board_;
    Board *next_board_;

    bool (*activation_func_)(bool, int);

};

#endif //GAME_OF_LIFE_GAMEOFLIFE_H
```

6.15 main.cc File Reference

```
#include "SimpleConsoleRenderer.h"
#include "GameOfLife.h"
```

Include dependency graph for main.cc:



Functions

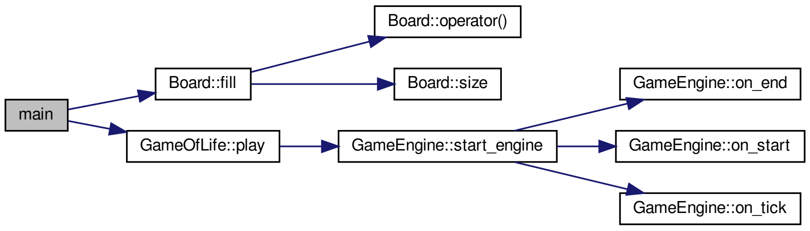
- int main ()

6.15.1 Function Documentation

6.15.1.1 main()

`int main ( )`

Definition at line 4 of file main.cc.
Here is the call graph for this function:



6.16 main.cc

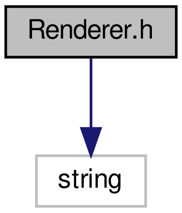
```
#include "SimpleConsoleRenderer.h"
#include "GameOfLife.h"

int main() {
    const unsigned size_x = 10;
    const unsigned size_y = 10;

    GameEngine::Config config{};

    config.framerate = 1;
    config.renderer = new SimpleConsoleRenderer;

    Board board(size_x, size_y);

    // Make the board empty
    board.fill(false);

    // Setup some config on the board
    board(1, 1) = true;
    board(1, 2) = true;
    board(1, 3) = true;

    board(3, 1) = true;
    board(3, 2) = true;
    board(3, 3) = true;

    board(6, 1) = true;
    board(6, 2) = true;
    board(6, 3) = true;

    board(9, 3) = true;
    board(9, 4) = true;
    board(9, 5) = true;

    // Start the GameEngine
    GameOfLife game_of_life(board, config);
    game_of_life.play();
}
```

6.17 README.md File Reference

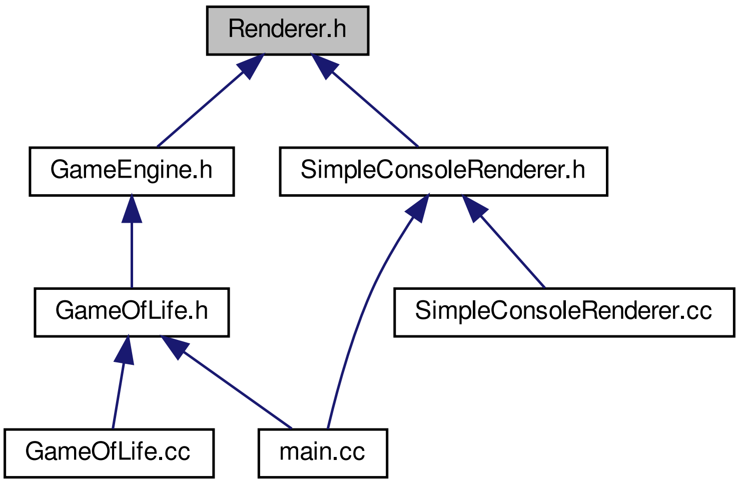
6.18 Renderer.h File Reference

`#include <string>`

Include dependency graph for Renderer.h:



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



Classes

- struct Coord

  Struct containing coordinates of different objects.
- class Renderer

  Basic base class for all renderers.

Enumerations

- enum class Color {
  Red , Green , Blue , Black ,
  White }

  Colors used in renderer.

6.18.1 Enumeration Type Documentation

6.18.1.1 Color

`enum Color  [strong]`

Colors used in renderer.

Enumerator

Red	
Green	
Blue	
Black	
White	

Definition at line 20 of file Renderer.h.

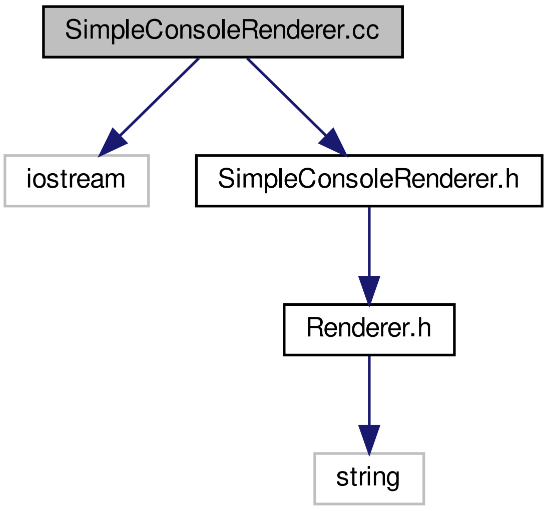
6.19 Renderer.h

```
//
// Created by mateu on 3/21/2021.
//

#ifndef GAME_OF_LIFE_RENDERER_H
#define GAME_OF_LIFE_RENDERER_H

#include <string>

struct Coord {
    Coord(int x_in, int y_in) :
            x(x_in),
            y(y_in) {}
    int x;
    int y;
};

enum class Color {
    Red,
    Green,
    Blue,
    Black,
    White
};

class Renderer {
public:

    virtual void create_window(int size_x, int size_y) = 0;

    virtual void draw_square(const Coord &position, int size_x, int size_y, const Color &fill) = 0;

    virtual void set_pixel(const Coord &position, const Color &fill) = 0;

    virtual void clear_screen(const Color &fill) = 0;

    virtual void show_text_big(const Coord &position, const std::string &text) = 0;

    virtual void show_text_medium(const Coord &position, const std::string &text) = 0;

    virtual void show_text_small(const Coord &position, const std::string &text) = 0;

    virtual void render() = 0;

    virtual ~Renderer() = default;

protected:
    int width_;
    int height_;

};


#endif //GAME_OF_LIFE_RENDERER_H
```

6.20 SimpleConsoleRenderer.cc File Reference

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

Include dependency graph for SimpleConsoleRenderer.cc:



Macros

- #define IS_ALPHA_NUMERIC(x) (x < 256)
- #define COLOR_BLACK 301
- #define COLOR_WHITE 300

6.20.1 Macro Definition Documentation

6.20.1.1 COLOR_BLACK

```
#define COLOR_BLACK 301
```

Definition at line 16 of file SimpleConsoleRenderer.cc.

6.20.1.2 COLOR_WHITE

```
#define COLOR_WHITE 300
```

Definition at line 17 of file SimpleConsoleRenderer.cc.

6.20.1.3 IS_ALPHA_NUMERIC

```
#define IS_ALPHA_NUMERIC(
                x ) (x < 256)
```

Definition at line 15 of file SimpleConsoleRenderer.cc.

6.21 SimpleConsoleRenderer.cc

```
00001 //
00002 // Created by mateu on 4/11/2021.
```

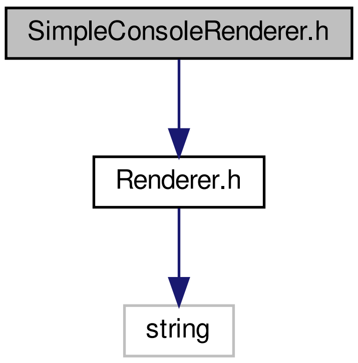
```
//

#include <iostream>
#include "SimpleConsoleRenderer.h"


#if defined(WIN32) || defined(_WIN32) || defined(__WIN32) && !defined(__CYGWIN__)

#include <Windows.h>

#endif

#define IS_ALPHA_NUMERIC(x) (x < 256)
#define COLOR_BLACK 301
#define COLOR_WHITE 300


void SimpleConsoleRenderer::create_window(int size_x, int size_y) {
    width_ = size_x;
    height_ = size_y;
    video_buffer_ = new GrayscalePixel[width_ * height_];
}

void SimpleConsoleRenderer::draw_square(const Coord &position, int size_x, int size_y, const Color
 &fill) {
    throw std::exception("Not implemented");
}

void SimpleConsoleRenderer::clear_screen(const Color &fill) {

    memset(video_buffer_, color_to_pixel(fill), width_ * height_);
    clear_window();
}

void SimpleConsoleRenderer::set_pixel(const Coord &position, const Color &fill) {
    video_buffer_[translate(position)] = color_to_pixel(fill);
}

void SimpleConsoleRenderer::show_text_big(const Coord &position, const std::string &text) {

    auto draw_stared_line = [this, &text, &position](int y) {
        for (int i = position.x; i < position.x + text.size() + 4; i++) {
            video_buffer_[translate({i, y})] = '*';
        }
    };

    draw_stared_line(position.y);

    int x = position.x;
    video_buffer_[translate({x++, position.y + 1})] = '*';
    video_buffer_[translate({x++, position.y + 1})] = ' ';
    for (int i = position.x + 1; i < position.x + text.size() + 1; i++) {
        video_buffer_[translate({i, position.y + 1})] = (unsigned char) text[i];
        x++;
    }
    video_buffer_[translate({x++, position.y + 1})] = ' ';
    video_buffer_[translate({x++, position.y + 1})] = '*';

    draw_stared_line(position.y + 2);
}

void SimpleConsoleRenderer::show_text_medium(const Coord &position, const std::string &text) {
    int x = position.x;
    video_buffer_[translate({x++, position.y})] = '*';
    video_buffer_[translate({x++, position.y})] = ' ';
    for (char i : text) {
        video_buffer_[translate({x++, position.y})] = (unsigned char) i;
    }
    video_buffer_[translate({x++, position.y})] = ' ';
    video_buffer_[translate({x++, position.y})] = '*';
}

void SimpleConsoleRenderer::show_text_small(const Coord &position, const std::string &text) {
    for (int i = position.x; i < position.x + text.size(); i++) {
        video_buffer_[translate({i, position.y})] = (unsigned char) text[i];
    }
}

void SimpleConsoleRenderer::render() {
    clear_window();
    for (int y = 0; y < height_; y++) {
      for (int x = 0; x < width_; x++) {

        auto item = video_buffer_[translate({x, y})];
        if (IS_ALPHA_NUMERIC(item))
          std::cout « (char)item;
        else if (item >= COLOR_BLACK)
```

```
          std::cout « ' ';
        else if (item == COLOR_WHITE)
          std::cout « "#";
        else
          throw std::exception("Bad value");

        std::cout « " ";
      }
      std::cout « std::endl;
    }
}

unsigned SimpleConsoleRenderer::translate(Coord position) {
    return position.y * width_ + position.x;
}

GrayscalePixel SimpleConsoleRenderer::color_to_pixel(const Color &color) {
    GrayscalePixel pixel_color = COLOR_BLACK;
    if (color == Color::White) {
        pixel_color = COLOR_WHITE;
    }
    return pixel_color;
}

void SimpleConsoleRenderer::clear_window() {
#if defined(WIN32) || defined(_WIN32) || defined(__WIN32) && !defined(__CYGWIN__)
    system("cls");
#else
    std::cout « "\x1B[2J\x1B[H";
#endif
}
```

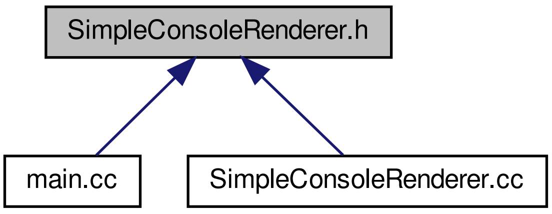
6.22 SimpleConsoleRenderer.h File Reference

`#include "Renderer.h"`

Include dependency graph for SimpleConsoleRenderer.h:



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



Classes

- class SimpleConsoleRenderer

Typedefs

- typedef uint16_t GrayscalePixel

6.22.1 Typedef Documentation

6.22.1.1 GrayscalePixel

```
typedef uint16_t GrayscalePixel
```

Definition at line 10 of file SimpleConsoleRenderer.h.

6.23 SimpleConsoleRenderer.h

```
//
// Created by mateu on 4/11/2021.
//

#ifndef GAME_OF_LIFE_SIMPLECONSOLERENDERER_H
#define GAME_OF_LIFE_SIMPLECONSOLERENDERER_H

#include "Renderer.h"

typedef uint16_t GrayscalePixel;

class SimpleConsoleRenderer : public Renderer {

    unsigned translate(Coord position);
    static GrayscalePixel color_to_pixel(const Color& color);

    void create_window(int size_x, int size_y) override;

    void draw_square(const Coord &position, int size_x, int size_y, const Color &fill) override;

    void clear_screen(const Color &fill) override;

    void set_pixel(const Coord &position, const Color &fill) override;

    void show_text_big(const Coord &position, const std::string &text) override;

    void show_text_medium(const Coord &position, const std::string &text) override;

    void show_text_small(const Coord &position, const std::string &text) override;

    void render() override;

    ~SimpleConsoleRenderer() override = default;

private:

    void clear_window();

    GrayscalePixel* video_buffer_;

};

#endif // GAME_OF_LIFE_SIMPLECONSOLERENDERER_H
```

Index