Game Of Life

Generated by Doxygen 1.8.13

Contents

Chapter 1

Main Page

Developers: Leyla Gasimova and Laman Mammadova

Date: 11th June, 2021

How to run the programm:

create folder named "build" (actually you can call it whatever, it is just a convention)
enter the folder
open the terminal and run the command:
cmake ..
if everything ran succesfully you can run next the command:
./GameOfLife
then input the size of the board you want meaning its columns and rows
the programm also will ask you which mode of the game you want to play clipped or circular
For clipped enter 1
For circular enter 2
Enjoy :)

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

File Documentation

3.1 build/CMakeFiles/3.10.2/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

3.1.1 Macro Definition Documentation

3.1.1.1 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

Definition at line 468 of file CMakeCCompilerId.c.

3.1.1.2 C_DIALECT

```
#define C_DIALECT
```

Definition at line 552 of file CMakeCCompilerId.c.

3.1.1.3 COMPILER_ID

```
#define COMPILER_ID ""
```

Definition at line 288 of file CMakeCCompilerId.c.

3.1.1.4 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 472 of file CMakeCCompilerId.c.

3.1.1.5 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 483 of file CMakeCCompilerId.c.

3.1.1.6 PLATFORM_ID

```
#define PLATFORM_ID
```

Definition at line 405 of file CMakeCCompilerId.c.

3.1.1.7 STRINGIFY

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

Definition at line 309 of file CMakeCCompilerId.c.

3.1.1.8 STRINGIFY_HELPER

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

Definition at line 308 of file CMakeCCompilerId.c.

3.1.2 Function Documentation

3.1.2.1 main()

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

Definition at line 572 of file CMakeCCompilerId.c.

3.1.3 Variable Documentation

3.1.3.1 info_arch

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

Definition at line 543 of file CMakeCCompilerId.c.

3.1.3.2 info_compiler

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

Definition at line 295 of file CMakeCCompilerId.c.

3.1.3.3 info_language_dialect_default

```
const char* info_language_dialect_default
```

Initial value:

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

Definition at line 561 of file CMakeCCompilerId.c.

3.1.3.4 info_platform

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

Definition at line 542 of file CMakeCCompilerId.c.

3.2 build/CMakeFiles/3.10.2/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

3.2.1 Macro Definition Documentation

3.2.1.1 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

Definition at line 453 of file CMakeCXXCompilerId.cpp.

3.2.1.2 COMPILER_ID

```
#define COMPILER_ID ""
```

Definition at line 273 of file CMakeCXXCompilerId.cpp.

3.2.1.3 CXX_STD

```
#define CXX_STD __cplusplus
```

Definition at line 536 of file CMakeCXXCompilerId.cpp.

3.2.1.4 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 457 of file CMakeCXXCompilerId.cpp.

3.2.1.5 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 468 of file CMakeCXXCompilerId.cpp.

3.2.1.6 PLATFORM_ID

```
#define PLATFORM_ID
```

Definition at line 390 of file CMakeCXXCompilerId.cpp.

3.2.1.7 STRINGIFY

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

Definition at line 294 of file CMakeCXXCompilerId.cpp.

3.2.1.8 STRINGIFY_HELPER

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

Definition at line 293 of file CMakeCXXCompilerId.cpp.

3.2.2 Function Documentation

3.2.2.1 main()

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

Definition at line 553 of file CMakeCXXCompilerId.cpp.

3.2.3 Variable Documentation

3.2.3.1 info_arch

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

Definition at line 528 of file CMakeCXXCompilerId.cpp.

3.2.3.2 info_compiler

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

Definition at line 280 of file CMakeCXXCompilerId.cpp.

3.2.3.3 info_language_dialect_default

```
const char* info_language_dialect_default
```

Initial value:

```
= "INFO" ":" "dialect_default["



  "98"
"]"
```

Definition at line 539 of file CMakeCXXCompilerId.cpp.

3.2.3.4 info_platform

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

Definition at line 527 of file CMakeCXXCompilerId.cpp.

3.3 build/CMakeFiles/feature_tests.c File Reference

Functions

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

Variables

- const char features [ ]

3.3.1 Function Documentation

3.3.1.1 main()

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

Definition at line 34 of file feature_tests.c.

3.3.2 Variable Documentation

3.3.2.1 features

```
const char features[]
```

Definition at line 2 of file feature_tests.c.

3.4 build/CMakeFiles/feature_tests.cxx File Reference

Functions

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

Variables

- const char features [ ]

3.4.1 Function Documentation

3.4.1.1 main()

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

Definition at line 405 of file feature_tests.cxx.

3.4.2 Variable Documentation

3.4.2.1 features

```
const char features[]
```

Definition at line 2 of file feature_tests.cxx.

3.5 Console/ainsi_console.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "board_drawer.h"
```

Functions

- void print (int **array, int rows, int columns)

3.5.1 Function Documentation

3.5.1.1 print()

```
void print (
            int ** array,
            int rows,
            int columns )
```

Given a double pointer to 2Darray, prints in terminal and checks for each element if it is dead or alive (0 or 1) based on that changes the color of the cell, blue for alive and white for dead, also clears the terminal before drawing the renewed board

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 5 of file ainsi_console.c.

3.6 Console/board_drawer.h File Reference

Macros

- #define BLK "\e[0;30m"

  AINSI code to change the color of text to black.
- #define RED "\e[0;31m"

  AINSI code to change the color of text to red.
- #define GRN "\e[0;32m"

  AINSI code to change the color of text to green.
- #define YEL "\e[0;33m"

  AINSI code to change the color of text to yellow.
- #define BLU "\e[0;34m"

  AINSI code to change the color of text to blue.
- #define MAG "\e[0;35m"

  AINSI code to change the color of text to magenta.
- #define CYN "\e[0;36m"

  AINSI code to change the color of text to cyan.
- #define WHT "\e[0;37m"

  AINSI code to change the color of text to white.
- #define BBLK "\e[1;30m"

  AINSI code to change text to bold black.
- #define BRED "\e[1;31m"

  AINSI code to change text to bold red.
- #define BGRN "\e[1;32m"

  AINSI code to change text to bold green.
- #define BYEL "\e[1;33m"

  AINSI code to change text to bold yellow.
- #define BBLU "\e[1;34m"

  AINSI code to change text to bold blue.
- #define BMAG "\e[1;35m"

  AINSI code to change text to bold magenta.
- #define BCYN "\e[1;36m"

  AINSI code to change text to bold cyan.
- #define BWHT "\e[1;37m"

  AINSI code to change text to bold white.
- #define UBLK "\e[4;30m"

  AINSI code to underline text with black.
- #define URED "\e[4;31m"

  AINSI code to underline text with red.

- #define UGRN "\e[4;32m"

  ANSI code to underline text with green.
- #define UYEL "\e[4;33m"

  ANSI code to underline text with yellow.
- #define UBLU "\e[4;34m"

  ANSI code to underline text with blue.
- #define UMAG "\e[4;35m"

  ANSI code to underline text with magenta.
- #define UCYN "\e[4;36m"

  ANSI code to underline text with cyan.
- #define UWHT "\e[4;37m"

  ANSI code to underline text with white.
- #define BLKB "\e[40m"

  ANSI code to change the background to black.
- #define REDB "\e[41m"

  ANSI code to change the background to red.
- #define GRNB "\e[42m"

  ANSI code to change the background to green.
- #define YELB "\e[43m"

  ANSI code to change the background to yellow.
- #define BLUB "\e[44m"

  ANSI code to change the background to blue.
- #define MAGB "\e[45m"

  ANSI code to change the background to magenta.
- #define CYNB "\e[46m"

  ANSI code to change the background to cyan.
- #define WHTB "\e[47m"

  ANSI code to change the background to white.
- #define BLKHB "\e[0;100m"

  ANSI code to change background to bright black.
- #define REDHB "\e[0;101m"

  ANSI code to change background to bright red.
- #define GRNHB "\e[0;102m"

  ANSI code to change background to bright green.
- #define YELHB "\e[0;103m"

  ANSI code to change background to bright yellow.
- #define BLUHB "\e[0;104m"

  ANSI code to change background to bright blue.
- #define MAGHB "\e[0;105m"

  ANSI code to change background to bright magenta.
- #define CYNHB "\e[0;106m"

  ANSI code to change background to bright cyan.
- #define WHTHB "\e[0;107m"

  ANSI code to change background to bright white.
- #define HBLK "\e[0;90m"

  ANSI code to change text to bright black.
- #define HRED "\e[0;91m"

  ANSI code to change text to bright red.
- #define HGRN "\e[0;92m"

  ANSI code to change text to bright green.
- #define HYEL "\e[0;93m"

AINSI code to change text to bright yellow.

- #define HBLU "\e[0;94m"

  AINSI code to change text to bright blue.
- #define HMAG "\e[0;95m"

  AINSI code to change text to bright magenta.
- #define HCYN "\e[0;96m"

  AINSI code to change text to bright cyan.
- #define HWHT "\e[0;97m"

  AINSI code to change text to bright white.
- #define BHBLK "\e[1;90m"

  AINSI code to change text to bright bold black.
- #define BHRED "\e[1;91m"

  AINSI code to change text to bright bold red.
- #define BHGRN "\e[1;92m"

  AINSI code to change text to bright bold green.
- #define BHYEL "\e[1;93m"

  AINSI code to change text to bright bold yellow.
- #define BHBLU "\e[1;94m"

  AINSI code to change text to bright bold blue.
- #define BHMAG "\e[1;95m"

  AINSI code to change text to bright bold magenta.
- #define BHCYN "\e[1;96m"

  AINSI code to change text to bright bold cyan.
- #define BHWHT "\e[1;97m"

  AINSI code to change text to bright bold white.
- #define reset "\e[0m"

  AINSI code to go back to regular text.
- #define clearscreen "\e[1;1H\e[2J"

  AINSI code to clear the terminal.

Functions

- void print (int **array, int rows, int columns)

3.6.1 Macro Definition Documentation

3.6.1.1 BBLK

```
#define BBLK "\e[1;30m"
```

AINSI code to change text to bold black.

Definition at line 23 of file board_drawer.h.

3.6.1.2 BBLU

```
#define BBLU "\e[1;34m"
```

AINSI code to change text to bold blue.

Definition at line 27 of file board_drawer.h.

3.6.1.3 BCYN

```
#define BCYN "\e[1;36m"
```

AINSI code to change text to bold cyan.

Definition at line 29 of file board_drawer.h.

3.6.1.4 BGRN

```
#define BGRN "\e[1;32m"
```

AINSI code to change text to bold green.

Definition at line 25 of file board_drawer.h.

3.6.1.5 BHBLK

```
#define BHBLK "\e[1;90m"
```

AINSI code to change text to bright bold black.

Definition at line 73 of file board_drawer.h.

3.6.1.6 BHBLU

```
#define BHBLU "\e[1;94m"
```

AINSI code to change text to bright bold blue.

Definition at line 77 of file board_drawer.h.

3.6.1.7 BHCYN

```
#define BHCYN "\e[1;96m"
```

AINSI code to change text to bright bold cyan.

Definition at line 79 of file board_drawer.h.

3.6.1.8 BHGRN

```
#define BHGRN "\e[1;92m"
```

AINSI code to change text to bright bold green.

Definition at line 75 of file board_drawer.h.

3.6.1.9 BHMAG

```
#define BHMAG "\e[1;95m"
```

AINSI code to change text to bright bold magenta.

Definition at line 78 of file board_drawer.h.

3.6.1.10 BHRED

```
#define BHRED "\e[1;91m"
```

AINSI code to change text to bright bold red.

Definition at line 74 of file board_drawer.h.

3.6.1.11 BHWHT

```
#define BHWHT "\e[1;97m"
```

AINSI code to change text to bright bold white.

Definition at line 80 of file board_drawer.h.

3.6.1.12 BHYEL

```
#define BHYEL "\e[1;93m"
```

AINSI code to change text to bright bold yellow.

Definition at line 76 of file board_drawer.h.

3.6.1.13 BLK

```
#define BLK "\e[0;30m"
```

AINSI code to change the color of text to black.

Definition at line 13 of file board_drawer.h.

3.6.1.14 BLKB

```
#define BLKB "\e[40m"
```

AINSI code to change the background to black.

Definition at line 43 of file board_drawer.h.

3.6.1.15 BLKHB

```
#define BLKHB "\e[0;100m"
```

AINSI code to change background to bright black.

Definition at line 53 of file board_drawer.h.

3.6.1.16 BLU

```
#define BLU "\e[0;34m"
```

AINSI code to change the color of text to blue.

Definition at line 17 of file board_drawer.h.

3.6.1.17 BLUB

```
#define BLUB "\e[44m"
```

AINSI code to change the background to blue.

Definition at line 47 of file board_drawer.h.

3.6.1.18 BLUHB

```
#define BLUHB "\e[0;104m"
```

AINSI code to change background to bright blue.

Definition at line 57 of file board_drawer.h.

3.6.1.19 BMAG

```
#define BMAG "\e[1;35m"
```

AINSI code to change text to bold magenta.

Definition at line 28 of file board_drawer.h.

3.6.1.20 BRED

```
#define BRED "\e[1;31m"
```

AINSI code to change text to bold red.

Definition at line 24 of file board_drawer.h.

3.6.1.21 BWHT

```
#define BWHT "\e[1;37m"
```

AINSI code to change text to bold white.

Definition at line 30 of file board_drawer.h.

3.6.1.22 BYEL

```
#define BYEL "\e[1;33m"
```

AINSI code to change text to bold yellow.

Definition at line 26 of file board_drawer.h.

3.6.1.23 clearscreen

```
#define clearscreen "\e[1;1H\e[2J"
```

AINSI code to clear the terminal.

Definition at line 84 of file board_drawer.h.

3.6.1.24 CYN

```
#define CYN "\e[0;36m"
```

AINSI code to change the color of text to cyan.

Definition at line 19 of file board_drawer.h.

3.6.1.25 CYNB

```
#define CYNB "\e[46m"
```

AINSI code to change the background to cyan.

Definition at line 49 of file board_drawer.h.

3.6.1.26 CYNHB

```
#define CYNHB "\e[0;106m"
```

AINSI code to change background to bright cyan.

Definition at line 59 of file board_drawer.h.

3.6.1.27 GRN

```
#define GRN "\e[0;32m"
```

AINSI code to change the color of text to green.

Definition at line 15 of file board_drawer.h.

3.6.1.28 GRNB

```
#define GRNB "\e[42m"
```

AINSI code to change the background to green.

Definition at line 45 of file board_drawer.h.

3.6.1.29 GRNHB

```
#define GRNHB "\e[0;102m"
```

AINSI code to change background to bright green.

Definition at line 55 of file board_drawer.h.

3.6.1.30 HBLK

```
#define HBLK "\e[0;90m"
```

AINSI code to change text to bright black.

Definition at line 63 of file board_drawer.h.

3.6.1.31 HBLU

```
#define HBLU "\e[0;94m"
```

AINSI code to change text to bright blue.

Definition at line 67 of file board_drawer.h.

3.6.1.32 HCYN

```
#define HCYN "\e[0;96m"
```

AINSI code to change text to bright cyan.

Definition at line 69 of file board_drawer.h.

3.6.1.33 HGRN

```
#define HGRN "\e[0;92m"
```

AINSI code to change text to bright green.

Definition at line 65 of file board_drawer.h.

3.6.1.34 HMAG

```
#define HMAG "\e[0;95m"
```

AINSI code to change text to bright magenta.

Definition at line 68 of file board_drawer.h.

3.6.1.35 HRED

```
#define HRED "\e[0;91m"
```

AINSI code to change text to bright red.

Definition at line 64 of file board_drawer.h.

3.6.1.36 HWHT

```
#define HWHT "\e[0;97m"
```

AINSI code to change text to bright white.

Definition at line 70 of file board_drawer.h.

3.6.1.37 HYEL

```
#define HYEL "\e[0;93m"
```

AINSI code to change text to bright yellow.

Definition at line 66 of file board_drawer.h.

3.6.1.38 MAG

```
#define MAG "\e[0;35m"
```

AINSI code to change the color of text to magenta.

Definition at line 18 of file board_drawer.h.

3.6.1.39 MAGB

```
#define MAGB "\e[45m"
```

AINSI code to change the background to magenta.

Definition at line 48 of file board_drawer.h.

3.6.1.40 MAGHB

```
#define MAGHB "\e[0;105m"
```

AINSI code to change background to bright magenta.

Definition at line 58 of file board_drawer.h.

3.6.1.41 RED

```
#define RED "\e[0;31m"
```

AINSI code to change the color of text to red.

Definition at line 14 of file board_drawer.h.

3.6.1.42 REDB

```
#define REDB "\e[41m"
```

AINSI code to change the background to red.

Definition at line 44 of file board_drawer.h.

3.6.1.43 REDHB

```
#define REDHB "\e[0;101m"
```

AINSI code to change background to bright red.

Definition at line 54 of file board_drawer.h.

3.6.1.44 reset

```
#define reset "\e[0m"
```

AINSI code to go back to regular text.

Definition at line 83 of file board_drawer.h.

3.6.1.45 UBLK

```
#define UBLK "\e[4;30m"
```

AINSI code to underline text with black.

Definition at line 33 of file board_drawer.h.

3.6.1.46 UBLU

```
#define UBLU "\e[4;34m"
```

AINSI code to underline text with blue.

Definition at line 37 of file board_drawer.h.

3.6.1.47 UCYN

```
#define UCYN "\e[4;36m"
```

AINSI code to underline text with cyan.

Definition at line 39 of file board_drawer.h.

3.6.1.48 UGRN

```
#define UGRN "\e[4;32m"
```

AINSI code to underline text with green.

Definition at line 35 of file board_drawer.h.

3.6.1.49 UMAG

```
#define UMAG "\e[4;35m"
```

AINSI code to underline text with magenta.

Definition at line 38 of file board_drawer.h.

3.6.1.50 URED

```
#define URED "\e[4;31m"
```

AINSI code to underline text with red.

Definition at line 34 of file board_drawer.h.

3.6.1.51 UWHT

```
#define UWHT "\e[4;37m"
```

AINSI code to underline text with white.

Definition at line 40 of file board_drawer.h.

3.6.1.52 UYEL

```
#define UYEL "\e[4;33m"
```

AINSI code to underline text with yellow.

Definition at line 36 of file board_drawer.h.

3.6.1.53 WHT

```
#define WHT "\e[0;37m"
```

AINSI code to change the color of text to white.

Definition at line 20 of file board_drawer.h.

3.6.1.54 WHTB

```
#define WHTB "\e[47m"
```

AINSI code to change the background to white.

Definition at line 50 of file board_drawer.h.

3.6.1.55 WHTHB

```
#define WHTHB "\e[0;107m"
```

AINSI code to change background to bright white.

Definition at line 60 of file board_drawer.h.

3.6.1.56 YEL

```
#define YEL "\e[0;33m"
```

AINSI code to change the color of text to yellow.

Definition at line 16 of file board_drawer.h.

3.6.1.57 YELB

```
#define YELB "\e[43m"
```

AINSI code to change the background to yellow.

Definition at line 46 of file board_drawer.h.

3.6.1.58 YELHB

```
#define YELHB "\e[0;103m"
```

AINSI code to change background to bright yellow.

Definition at line 56 of file board_drawer.h.

3.6.2 Function Documentation

3.6.2.1 print()

```
void print (
            int ** array,
            int rows,
            int columns )
```

Given a double pointer to 2Darray, prints in terminal and checks for each element if it is dead or alive (0 or 1) based on that changes the color of the cell, blue for alive and white for dead, also clears the terminal before drawing the renewed board

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 5 of file ainsi_console.c.

3.7 Game/life.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "life.h"
```

Functions

- int ** create_2d_array (int rows, int columns)
- void initialize (int **array, int rows, int columns)
- void random_alive (int **array, int rows, int columns)
- void check_alive_or_no_circular (int **array, int rows, int columns, int r, int c, int **array_copy)
- void circular (int **array, int rows, int columns)
- void check_alive_or_no_clipped (int **array, int rows, int columns, int r, int c, int **array_copy)
- void clipped (int **array, int rows, int columns)

3.7.1 Function Documentation

3.7.1.1 check_alive_or_no_circular()

```
void check_alive_or_no_circular (
            int ** array,
            int rows,
            int columns,
            int r,
            int c,
            int ** array_copy )
```

Definition at line 37 of file life.c.

3.7.1.2 check_alive_or_no_clipped()

```
void check_alive_or_no_clipped (
            int ** array,
            int rows,
            int columns,
            int r,
            int c,
            int ** array_copy )
```

Takes a double pointer to a 2D array, the number of rows and of columns, index for row and column of specific element and double pointer to the copy of the array The function counts the alive neighbor cells and based on that deducts if the specified cell is alive or dead and changes it in the copy of the array, not in the original

Parameters

in	*array*	
in	*rows*	
in	*columns*	
in	*r*	
in	*c*	
in	*copy_array*	

Definition at line 132 of file life.c.

3.7.1.3 circular()

```
void circular (
            int ** array,
            int rows,
            int columns )
```

The mode for circular game of life, when we do not consider boundaries Given a double pointer to the 2Darray creates its copy, then goes into for loop and checks for each cell if it is alive or no using check_alive_or_no function updates the copy of the array then updates the original and frees memory from copy

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 104 of file life.c.

3.7.1.4 clipped()

```
void clipped (
            int ** array,
            int rows,
            int columns )
```

The mode for clipped game of life, when we consider boundaries Given a double pointer to the 2Darray creates its copy but which has 2 more rows and columns, and extra rows and columns are initialized to zero (dead cells) then goes into for loop and checks for each cell of the bigger copy if it is alive or no using check_alive_or_no_clipped function updates the bigger copy of the array then updates the original and frees memory from copy

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 167 of file life.c.

3.7.1.5 create_2d_array()

```
int** create_2d_array (
            int rows,
            int columns )
```

Allocates 2d array given the number of rows and columns, return double pointer to the array.

Parameters

in	*rows*	
in	*columns*	
out	*pointer_to_2Darray*	

Definition at line 6 of file life.c.

3.7.1.6 initialize()

```
void initialize (
             int ** array,
             int rows,
             int columns )
```

Takes a double pointer to a 2D array, the number of rows and of columns and initializes all elements of the array to 0

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 14 of file life.c.

3.7.1.7 random_alive()

```
void random_alive (
             int ** array,
             int rows,
             int columns )
```

Takes a double pointer to a 2D array, the number of rows and of columns and randomly assigns 0 or 1 to all elements

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 25 of file life.c.

3.8 Game/life.h File Reference

Functions

- int ** create_2d_array (int rows, int columns)
- void initialize (int **array, int rows, int columns)
- void random_alive (int **array, int rows, int columns)
- void check_alive_or_no (int **array, int rows, int columns, int r, int c, int **copy_array)
- void circular (int **array, int rows, int columns)
- void check_alive_or_no_clipped (int **array, int rows, int columns, int r, int c, int **array_copy)
- void clipped (int **array, int rows, int columns)

3.8.1 Function Documentation

3.8.1.1 check_alive_or_no()

```
void check_alive_or_no (
            int ** array,
            int rows,
            int columns,
            int r,
            int c,
            int ** copy_array )
```

Takes a double pointer to a 2D array, the number of rows and of columns, index for row and column of specific element and double pointer to the copy of the array The function counts the alive neighbor cells and based on that deducts if the specified cell is alive or dead and changes it in the copy of the array, not in the original

Parameters

in	*array*	
in	*rows*	
in	*columns*	
in	*r*	
in	*c*	
in	*copy_array*	

3.8.1.2 check_alive_or_no_clipped()

```
void check_alive_or_no_clipped (
            int ** array,
            int rows,
            int columns,
            int r,
```

```
            int c,
            int ** array_copy )
```

Takes a double pointer to a 2D array, the number of rows and of columns, index for row and column of specific element and double pointer to the copy of the array The function counts the alive neighbor cells and based on that deducts if the specified cell is alive or dead and changes it in the copy of the array, not in the original

Parameters

in	*array*	
in	*rows*	
in	*columns*	
in	*r*	
in	*c*	
in	*copy_array*	

Definition at line 132 of file life.c.

3.8.1.3 circular()

```
void circular (
            int ** array,
            int rows,
            int columns )
```

The mode for circular game of life, when we do not consider boundaries Given a double pointer to the 2Darray creates its copy, then goes into for loop and checks for each cell if it is alive or no using check_alive_or_no function updates the copy of the array then updates the original and frees memory from copy

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 104 of file life.c.

3.8.1.4 clipped()

```
void clipped (
            int ** array,
            int rows,
            int columns )
```

The mode for clipped game of life, when we consider boundaries Given a double pointer to the 2Darray creates its copy but which has 2 more rows and columns, and extra rows and columns are initialized to zero (dead cells) then goes into for loop and checks for each cell of the bigger copy if it is alive or no using check_alive_or_no_clipped function updates the bigger copy of the array then updates the original and frees memory from copy

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 167 of file life.c.

3.8.1.5 create_2d_array()

```
int** create_2d_array (
            int rows,
            int columns )
```

Allocates 2d array given the number of rows and columns, return double pointer to the array.

Parameters

in	*rows*	
in	*columns*	
out	*pointer_to_2Darray*	

Definition at line 6 of file life.c.

3.8.1.6 initialize()

```
void initialize (
            int ** array,
            int rows,
            int columns )
```

Takes a double pointer to a 2D array, the number of rows and of columns and initializes all elements of the array to 0

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 14 of file life.c.

3.8.1.7 random_alive()

```
void random_alive (
            int ** array,
            int rows,
            int columns )
```

Takes a double pointer to a 2D array, the number of rows and of columns and randomly assigns 0 or 1 to all elements

Parameters

in	*array*	
in	*rows*	
in	*columns*	

Definition at line 25 of file life.c.

3.9 main.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include "./Game/life.h"
#include "./Console/board_drawer.h"
```

Functions

- int main ()

3.9.1 Function Documentation

3.9.1.1 main()

```
int main ( )
```

Definition at line 28 of file main.c.

Index