Conway

Generated by Doxygen 1.9.1

Chapter 1

readme

I. Project Purpose

This project aims to implement Conway's Game of Life, providing a visual interface that allows users to intuitively observe the game's operation. It also supports loading specific patterns, saving game states as images, and other features to enhance the game's playability and entertainment value.

II. Functional Introduction

1.Random Pattern Generation: The program can randomly generate the initial cell states, with each cell having a 50% chance of being alive.

2.Loading Pattern Files: It supports loading specific cell patterns from RLE files and placing them at designated locations.

3.Visual Interface: A text-based user interface is created using the ncurses library, which displays the cell states in real-time, with live cells represented by '∗' and dead cells by spaces.

4.Saving Game State as Image: The current game state can be saved as a BMP image file, with live cells in white and dead cells in black. The filename is generated based on the current time.

5.Toroidal Boundary Mode: The game supports toroidal boundary mode, meaning the world is wrap-around, and cells on the edges will "wrap" to the opposite side.

6.Real-time Updates and Control: The game updates cell states in real-time, and users can control the game through key presses. Pressing 'q' exits the program, and pressing 's' saves the current state as an image.

III. Usage

1.Compile the Project: Run the following command in the project root directory: make.

2.Run the Program: After compilation, run the program by specifying the width and height of the game area. You can also load a specific pattern file; otherwise, it will generate a random pattern by default.

Command: ./main <width> <height> [rle_file]

3.Generate Doxygen Documentation: Run the following command to generate Doxygen documentation: make doc. This will create a doc folder containing html and latex subfolders. You can view the documentation in html/index.html or latex/refman.pdf.

4.Clean Generated Files: Run make clean to remove all generated files.

IV. Dependencies

The function definitions are placed in the Conway.h file, and the function implementations are in the Conway.↩ c file. The main function, test programs, and ncurses environment are in the main.c file. The ncurses library is used to create the text-based user interface, providing screen drawing and keyboard input handling functions. The stb_image_write.h is used to save image files, supporting the saving of game states in BMP format.

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 World Struct Reference

Represents the state of the game world.

`#include <Conway.h>`

Public Attributes

- int width

  Width of the matrix.
- int height

  Height of the matrix.
- int ** cell

  Two-dimensional array, where each element represents a cell. 0 means dead, 1 means alive.

4.1.1 Detailed Description

Represents the state of the game world.

This structure contains the width, height, and a two-dimensional array representing the state of the cells.

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

- Conway.h

Chapter 5

File Documentation

5.1 Conway.h File Reference

Contains the data structures and function declarations for Conway's Game of Life.

Classes

- struct World

  Represents the state of the game world.

Functions

- void init_cells (int _w, int _h)

  Initializes the cell states.
- void set_rand_cells ()

  Randomly generates cell states.
- void load_pattern (int _x, int _y, char *rle_file)

  Loads a pattern file.
- void draw_cells ()

  Draws the cell states.
- int count_neighbors (int x, int y)

  Counts the number of neighbors for a cell.
- void update_cells ()

  Updates the cell states.
- void free_cells ()

  Frees the memory allocated for the cell states.
- void save_image (int width, int height, int **cell)

  Saves the current cell state as an image.

Variables

- struct World world

  The global instance of the game world.
- int boundary

  Controls the boundary mode of the world.

5.1.1 Detailed Description

Contains the data structures and function declarations for Conway's Game of Life.

Date

2025-03-14 This header file defines the data structure for the game world and the declarations of related functions.

5.1.2 Function Documentation

5.1.2.1 count_neighbors()

```
int count_neighbors (
            int x,
            int y )
```

Counts the number of neighbors for a cell.

Parameters

x	The x-coordinate of the cell.
y	The y-coordinate of the cell.

Returns

The number of neighbors for the cell.

Calculates the number of neighbors for a cell based on the boundary mode (fixed or toroidal).

5.1.2.2 draw_cells()

```
void draw_cells ( )
```

Draws the cell states.

Uses the ncurses library to draw the cell states on the screen, with live cells represented by '$*$' and dead cells by ' '.

5.1.2.3 free_cells()

```
void free_cells ( )
```

Frees the memory allocated for the cell states.

Releases the memory allocated for the cell states.

5.1.2.4 init_cells()

```
void init_cells (
            int _w,
            int _h )
```

Initializes the cell states.

Parameters

↩ _↩ *w*	The width of the game area.
↩ _↩ *h*	The height of the game area.

Allocates memory for the game area and initializes all cells to the dead state.

5.1.2.5 load_pattern()

```
void load_pattern (
            int _x,
            int _y,
            char * rle_file )
```

Loads a pattern file.

Parameters

_x	The starting x-coordinate for the pattern.
_y	The starting y-coordinate for the pattern.
rle_file	The path to the pattern file.

Loads a cell pattern from an RLE (Run-Length Encoded) file and places it at the specified location.

5.1.2.6 save_image()

```
void save_image (
            int width,
            int height,
            int ** cell )
```

Saves the current cell state as an image.

Parameters

width	The width of the game area.
height	The height of the game area.
cell	The two-dimensional array representing the cell states.

Saves the current cell state as a BMP image, with live cells represented by white and dead cells by black. The image filename is generated based on the current time, in the format "image_YYYYMMDD_HHMMSS.bmp".

5.1.2.7 set_rand_cells()

```
void set_rand_cells ( )
```

Randomly generates cell states.

Randomly generates the initial state of the cells, with each cell having a 50% chance of being alive.

5.1.2.8 update_cells()

```
void update_cells ( )
```

Updates the cell states.

Updates the cell states according to the rules of Conway's Game of Life.

5.1.3 Variable Documentation

5.1.3.1 boundary

```
int boundary  [extern]
```

Controls the boundary mode of the world.

0 means the outside of the world is empty; 1 means the world is toroidal (wraps around).

5.1.3.2 world

```
struct World world  [extern]
```

The global instance of the game world.

This global variable represents the current state of the game world.

Index