

## Game of Life

Generated by Doxygen 1.10.0



<b>1 README</b>	<b>1</b>
<b>2 Topic Index</b>	<b>3</b>
2.1 Topics . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 File Index</b>	<b>7</b>
4.1 File List . . . . .	7
<b>5 Topic Documentation</b>	<b>9</b>
5.1 kirjastot . . . . .	9
5.2 Määrittelyt ja vakiot . . . . .	9
5.2.1 Detailed Description . . . . .	9
5.2.2 Macro Definition Documentation . . . . .	10
5.2.2.1 COLUMNS . . . . .	10
5.2.2.2 Generations . . . . .	10
5.2.2.3 Gray . . . . .	10
5.2.2.4 ground . . . . .	10
5.2.2.5 Human_pair . . . . .	10
5.2.2.6 Mountain . . . . .	10
5.2.2.7 Orc_pair . . . . .	10
5.2.2.8 River . . . . .	11
5.2.2.9 ROWS . . . . .	11
5.3 Funktioiden prototyypit . . . . .	11
5.3.1 Detailed Description . . . . .	11
5.3.2 Function Documentation . . . . .	11
5.3.2.1 calculateSim() . . . . .	11
5.3.2.2 game_init() . . . . .	12
5.3.2.3 printBoard() . . . . .	12
5.3.2.4 updateBoard() . . . . .	12
<b>6 Class Documentation</b>	<b>15</b>
6.1 cell Struct Reference . . . . .	15
6.1.1 Detailed Description . . . . .	15
6.1.2 Member Data Documentation . . . . .	15
6.1.2.1 current . . . . .	15
6.1.2.2 future . . . . .	15
6.1.2.3 type . . . . .	15
<b>7 File Documentation</b>	<b>17</b>
7.1 functions.c File Reference . . . . .	17
7.1.1 Detailed Description . . . . .	17

7.1.2 Function Documentation . . . . .	17
7.1.2.1 countNeighbors() . . . . .	17
7.2 header.h File Reference . . . . .	18
7.2.1 Detailed Description . . . . .	19
7.3 header.h . . . . .	19
<b>Index</b>	<b>21</b>

# Chapter 1

## README

### Game of Life

Tämä ohjelma toteuttaa Conwayn Game of Life -pelin kaltaisen taistelun kahden eri armeijan välillä.

Game of Life on soluautomaatti, joka toimii kaksiulotteisella ruudukolla. Peli etenee askel askeleelta sääntöjen mukaisesti, jotka määrittävät, mitkä solut pysyvät elossa, syntyvät uudestaan tai kuolevat. Ohjelma käyttää tekstipohjaista käyttöliittymää näyttääkseen pelin tilan ja päivittääkseen sitä jokaisen askeleen jälkeen.

Sisällytetyt kirjastot Ohjelma käyttää seuraavia kirjastoja:

stdio.h: Perustoimintoja varten. ncurses.h: Visuaalinen käyttöliittymä. unistd.h: usleep-funktio, joka mahdollistaa millisekuntien viivästykset.



## Chapter 2

# Topic Index

### 2.1 Topics

Here is a list of all topics with brief descriptions:

kirjastot . . . . .	9
Määrittelyt ja vakiot . . . . .	9
Funktioiden prototyypit . . . . .	11





## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">cell</a>	Pelilaudan solu, joka tallentaa nykyisen ja tulevan tilan sekä tyyppin . . . . .	<a href="#">15</a>
----------------------	--	--------------------



## Chapter 4

# File Index

### 4.1 File List

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

<a href="#">functions.c</a>	Tämä tiedosto sisältää ohjelman tarvittavat funktiot . . . . .	17
<a href="#">header.h</a>	Tämä tiedosto sisältää ohjelman tarvittavat kirjastot, määrittelyt ja funktioiden prototyypit . . .	18



# Chapter 5

## Topic Documentation

### 5.1 kirjastot

Ohjelmassa tarvittavat kirjastot.

Ohjelmassa tarvittavat kirjastot.

### 5.2 Määrittelyt ja vakiot

Ohjelmassa käytettävät määrittelyt ja vakiot.

#### Macros

- #define ROWS 52
- #define COLUMNS 52
- #define Generations
- #define ground 1
- #define Human\_pair 5
- #define River 2
- #define Mountain 3
- #define Gray 4
- #define Orc\_pair 6

#### 5.2.1 Detailed Description

Ohjelmassa käytettävät määrittelyt ja vakiot.

< Perustoimintoja varten. < Visuaalinen käyttöliittymä. < usleep-funktio, joka mahdollistaa millisekuntien viivästykset.

## 5.2.2 Macro Definition Documentation

### 5.2.2.1 COLUMNS

```
#define COLUMNS 52
```

Pelilaudan sarakkeiden määrä.

### 5.2.2.2 Generations

```
#define Generations
```

Generaatioiden määrä.

### 5.2.2.3 Gray

```
#define Gray 4
```

Harmaan väri.

### 5.2.2.4 ground

```
#define ground 1
```

Maan väri.

### 5.2.2.5 Human\_pair

```
#define Human_pair 5
```

Ihmisten väri.

### 5.2.2.6 Mountain

```
#define Mountain 3
```

Vuoren väri.

### 5.2.2.7 Orc\_pair

```
#define Orc_pair 6
```

Örkkien väri.

### 5.2.2.8 River

```
#define River 2
```

Joen väri.

### 5.2.2.9 ROWS

```
#define ROWS 52
```

Pelilaudan rivien määrä.

## 5.3 Funktioiden prototyypit

Ohjelmassa tarvittavat funktioiden prototyypit.

### Functions

- void `game_init` (struct `cell` board[`ROWS`][`COLUMNS`])
- void `countNeighbors` (struct `cell` board[`ROWS`][`COLUMNS`], int r, int c, int \*humanCount, int \*orcCount)
- void `printBoard` (WINDOW \*w, struct `cell` board[`ROWS`][`COLUMNS`])
- void `updateBoard` (struct `cell` board[`ROWS`][`COLUMNS`])
- void `calculateSim` (struct `cell` board[`ROWS`][`COLUMNS`])

### 5.3.1 Detailed Description

Ohjelmassa tarvittavat funktioiden prototyypit.

### 5.3.2 Function Documentation

#### 5.3.2.1 calculateSim()

```
void calculateSim (  
    struct cell board[ROWS][COLUMNS] )
```

Tämä funktio on osa simulointia, joka käsittelee pelilaudan solujen tilojen päivitystä pelin jokaisella askeleella. Jokaiselle solulle lasketaan ympäröivien solujen määrä, jotka ovat joko ihmisiä tai örkkejä, ja tarkistetaan myös, onko vieressä vihollissoluja.

Ensin käydään läpi jokainen solu pelilaudalla. Lasketaan ympäröivien ihmisten ja örkkien määrä kullekin solulle käyttämällä `countNeighbors`-funktia. Tarkistetaan, että solu ei ole este (vuori, vesi tai seinä). Tarkistetaan solun nykyinen tila ja sovelletaan sääntöjä sen perusteella:

Jos solu on ihminen: Jos naapurien määrä on alle 2 tai yli 3, ihminen kuolee alikuolleisuuden tai ylikuolleisuuden vuoksi. Muutoin yritetään liikkua oikealle, ja jos se on mahdollista, siirrytään sinne. Jos oikealla on örkki ja ei ole vierekkäisiä vihollissoluja, taistelu tapahtuu ja molemmat kuolevat.

Jos solu on örkki, sitä koskevat vastaavat säännöt ihmisten sijasta. Jos solu on tyhjä ja naapureiden joukossa on 3 ihmistä ja 0 örkkiä, syntyy uusi ihminen. Jos taas naapureiden joukossa on 3 örkkiä ja 0 ihmistä, syntyy uusi örkki.

Lopuksi päivitetään pelilauta tulevaisuuden tilan mukaiseksi.

## Parameters

<code>board</code>	Pelilauta, jonka solujen tiloja päivitetään.
--------------------	--

5.3.2.2 `game_init()`

```
void game_init (
    struct cell board[ROWS][COLUMNS] )
```

Lyhyt selitys ohjelman toiminnasta

Alustaa pelilaudan lataamalla tiedon game.txt-tiedostosta.

Funktion idea on aloittaa itse peli. Ensimmäisenä avaamalla game.txt tiedostossa sisältävän pelilaudan johon on "piirretty" Pelin kartta Aluksi ongelmien välttämiseksi structin current ja future laudat nollataan. Seuraavaksi avataan game.txt jossa käydään jokainen laudan solu läpi lukien, että onko solu 0 (tyhjä solu), 1 (Ihmisten solu), 2(Örkkien solu) (W) vesistö solu tai (^) vuori solu.

Lopulta funktio lopettaa sulkemalla avatun tiedoston ja täten pelilaudat on asetettu oikein tulevia funktioita varten

## Parameters

<code>board</code>	Pelilauta, johon tiedot ladataan.
--------------------	-----------------------------------

5.3.2.3 `printBoard()`

```
void printBoard (
    WINDOW * w,
    struct cell board[ROWS][COLUMNS] )
```

Lyhyt selitys ohjelman toiminnasta Näyttää pelilaudan nykyisen tilan näytöllä ja päivittää sen. Funktio vastaa pelilaudan nykyisen tilan näyttämisestä näytöllä. Se tyhjentää ensin ncurses-näytön, jonka jälkeen piirtää pelitapahtumat. Samalla se asettaa boolean-arvon "insideMountain" kertomaan, onko kohdattu vuoristoalue, jota varten vuoristoso-lut väritetään harmaaksi. Käy sitten läpi jokaisen pelilaudan solun seuraavasti:

--> Jos solu on jokisolu (W), piirretään joki-symboli näytölle. --> Jos solu on vuorisolu (^), vaihdetaan "insideMountain" -booleanin arvoa ja piirretään vuorisymboli näytölle. --> Jos "insideMountain" on tosi, solu väritetään harmaaksi. --> Jos solu on elävä ihmissolu (current == 1), piirretään ihmis-symboli näytölle. --> Jos solu on elävä örkkisolu (current == 2), piirretään örkki-symboli näytölle. --> Muussa tapauksessa solu piirretään tyhjänä näytölle.

Lopuksi funktio piirtää laatikon (box) ikkunan ympärille ja päivittää näytön (wrefresh).

## Parameters

<code>w</code>	NCurses-ikkuna, johon pelilauta piirretään.
<code>board</code>	Pelilauta, jossa solujen tilat sijaitsevat.

5.3.2.4 `updateBoard()`

```
void updateBoard (
```



```
struct cell board[ROWS][COLUMNS] )
```

Lyhyt selitys ohjelman toiminnasta

Päivittää pelilaudan nykyisen tilan tulevaisuuden tilaan. Funktio korvaa vanhemman taulukon future taululla joka on päivitetty alkuperäisestä calculateSim funktion laskujen mukaisesti

#### Parameters

<code>board</code>	Pelilauta, joka päivitetään.
--------------------	------------------------------



## Chapter 6

# Class Documentation

### 6.1 cell Struct Reference

Pelilaudan solu, joka tallentaa nykyisen ja tulevan tilan sekä tyypin.

```
#include <header.h>
```

#### Public Attributes

- int [current](#)
- int [future](#)
- char [type](#)

#### 6.1.1 Detailed Description

Pelilaudan solu, joka tallentaa nykyisen ja tulevan tilan sekä tyypin.

#### 6.1.2 Member Data Documentation

##### 6.1.2.1 current

```
int cell::current
```

Nykyinen tila.

##### 6.1.2.2 future

```
int cell::future
```

Tuleva tila.

##### 6.1.2.3 type

```
char cell::type
```

Solun tyyppi ('N' normaali, 'W' vesi/joki, '^' vuori).

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

- [header.h](#)



# Chapter 7

## File Documentation

### 7.1 functions.c File Reference

Tämä tiedosto sisältää ohjelman tarvittavat funktiot.

```
#include "header.h"
```

#### Functions

- void `game_init` (struct `cell` board[`ROWS`][`COLUMNS`])
- void `calculateSim` (struct `cell` board[`ROWS`][`COLUMNS`])
- void `countNeighbors` (struct `cell` board[`ROWS`][`COLUMNS`], int r, int c, int \*humanCount, int \*orcCount, bool \*adjacentEnemy)
- void `printBoard` (WINDOW \*w, struct `cell` board[`ROWS`][`COLUMNS`])
- void `updateBoard` (struct `cell` board[`ROWS`][`COLUMNS`])

#### 7.1.1 Detailed Description

Tämä tiedosto sisältää ohjelman tarvittavat funktiot.

#### 7.1.2 Function Documentation

##### 7.1.2.1 countNeighbors()

```
void countNeighbors (
    struct cell board[ROWS][COLUMNS],
    int r,
    int c,
    int * humanCount,
    int * orcCount,
    bool * adjacentEnemy )
```

Laskee tietyn solun ympärillä olevien naapurisolujen määrän ja tarkistaa, onko vieressä vihollissoluja. Tämä funktio laskee tietyn solun ympärillä olevien naapurisolujen määrän ja samalla tarkistaa, onko vieressä vihollissoluja.

Aluksi alustetaan muuttujat `humanCount`, `orcCount` ja `adjacentEnemy`. Käydään läpi solun ympärillä olevat solut käyttämällä kaksiulotteista silmukkaa. Tarkistetaan, että käsiteltävä solu ei ole itse solu, jota tarkastellaan, ja että se ei ole laudan reunalla tai este (vuori, vesi tai seinä). Jos naapuri on ihminen, lisätään `humanCount`-muuttujan arvoa yhdellä tai jos naapuri on örkki, lisätään `orcCount`-muuttujan arvoa yhdellä.

Tarkistetaan myös, onko naapurisolussa vihollissolu. Jos solun vieressä on vihollissolu, asetetaan `adjacentEnemy`-Booleani arvoksi `true`. Tämän toiminnan jälkeen `humanCount` ja `orcCount` sisältävät ihmis- ja örkinäkyvien solujen määrän, ja `adjacentEnemy` kertoo, onko tarkasteltavan solun vieressä vihollissoluja. Näitä tietoja käytetään edelleen pelin simulaatiossa, kun päätetään solun tuleva tila.

#### Parameters

<code>board</code>	Pelilauta, jossa solut sijaitsevat.
<code>r</code>	Solun rivi-indeksi.
<code>c</code>	Solun sarakkeen indeksi.
<code>humanCount</code>	Ihmisten määrä solun ympärillä (tulos).
<code>orcCount</code>	Örkkien määrä solun ympärillä (tulos).
<code>adjacentEnemy</code>	Onko solun vieressä vihollissoluja (tulos).

## 7.2 header.h File Reference

Tämä tiedosto sisältää ohjelman tarvittavat kirjastot, määrittelyt ja funktioiden prototyytit.

```
#include <stdio.h>
#include <ncurses.h>
#include <unistd.h>
```

#### Classes

- struct [cell](#)  
*Pelilaudan solu, joka tallentaa nykyisen ja tulevan tilan sekä tyypin.*

#### Macros

- #define [ROWS](#) 52
- #define [COLUMNS](#) 52
- #define [Generations](#)
- #define [ground](#) 1
- #define [Human\\_pair](#) 5
- #define [River](#) 2
- #define [Mountain](#) 3
- #define [Gray](#) 4
- #define [Orc\\_pair](#) 6

#### Functions

- void [game\\_init](#) (struct [cell](#) board[[ROWS](#)][[COLUMNS](#)])
- void [countNeighbors](#) (struct [cell](#) board[[ROWS](#)][[COLUMNS](#)], int r, int c, int \*humanCount, int \*orcCount)
- void [printBoard](#) (WINDOW \*w, struct [cell](#) board[[ROWS](#)][[COLUMNS](#)])
- void [updateBoard](#) (struct [cell](#) board[[ROWS](#)][[COLUMNS](#)])
- void [calculateSim](#) (struct [cell](#) board[[ROWS](#)][[COLUMNS](#)])

## 7.2.1 Detailed Description

Tämä tiedosto sisältää ohjelman tarvittavat kirjastot, määrittelyt ja funktioiden prototyypit.

## 7.3 header.h

[Go to the documentation of this file.](#)

```
00001 #ifndef HEADER_H
00002 #define HEADER_H
00003
00015 #include <stdio.h>
00016 #include <ncurses.h>
00017 #include <unistd.h>
00026 #define ROWS 52
00027 #define COLUMNS 52
00028 #define Generations
00029 #define ground 1
00030 #define Human_pair 5
00031 #define River 2
00032 #define Mountain 3
00033 #define Gray 4
00034 #define Orc_pair 6
00041 struct cell
00042 {
00043     int current;
00044     int future;
00045     char type;
00046 };
00047
00053 void game_init(struct cell board[ROWS][COLUMNS]);
00054 void countNeighbors(struct cell board[ROWS][COLUMNS], int r, int c, int* humanCount, int* orcCount);
00055 void printBoard(WINDOW *w, struct cell board[ROWS][COLUMNS]);
00056 void updateBoard(struct cell board[ROWS][COLUMNS]);
00057 void calculateSim(struct cell board[ROWS][COLUMNS]);
00060 #endif /* HEADER_H */
```





# Index

calculateSim  
    Funktioiden prototyytit, [11](#)  
cell, [15](#)  
    current, [15](#)  
    future, [15](#)  
    type, [15](#)  
COLUMNS  
    Määrittelyt ja vakiot, [10](#)  
countNeighbors  
    functions.c, [17](#)  
current  
    cell, [15](#)  
  
functions.c, [17](#)  
    countNeighbors, [17](#)  
Funktioiden prototyytit, [11](#)  
    calculateSim, [11](#)  
    game\_init, [12](#)  
    printBoard, [12](#)  
    updateBoard, [12](#)  
future  
    cell, [15](#)  
  
game\_init  
    Funktioiden prototyytit, [12](#)  
Generations  
    Määrittelyt ja vakiot, [10](#)  
Gray  
    Määrittelyt ja vakiot, [10](#)  
ground  
    Määrittelyt ja vakiot, [10](#)  
  
header.h, [18](#)  
Human\_pair  
    Määrittelyt ja vakiot, [10](#)  
  
kirjastot, [9](#)  
  
Mountain  
    Määrittelyt ja vakiot, [10](#)  
Määrittelyt ja vakiot, [9](#)  
    COLUMNS, [10](#)  
    Generations, [10](#)  
    Gray, [10](#)  
    ground, [10](#)  
    Human\_pair, [10](#)  
    Mountain, [10](#)  
    Orc\_pair, [10](#)  
    River, [10](#)  
    ROWS, [11](#)

Orc\_pair  
    Määrittelyt ja vakiot, [10](#)  
  
printBoard  
    Funktioiden prototyytit, [12](#)  
  
README, [1](#)  
River  
    Määrittelyt ja vakiot, [10](#)  
ROWS  
    Määrittelyt ja vakiot, [11](#)  
  
type  
    cell, [15](#)  
  
updateBoard  
    Funktioiden prototyytit, [12](#)