Game of Life

Készítette Doxygen 1.9.2

1. Specifikacio **1**

1.1. Mi fogadja a felhasználót a program megnyitásakor? . . . 1

1.1.1. 1. "Játék" . . . 1

1.1.2. 2. "Betöltés" . . . 2

1.1.3. 3. "Súgó" . . . 2

1.2. A mentett játékállások . . . 2

2. Adatszerkezet-mutató **3**

2.1. Adatszerkezetek . . . 3

3. Fájlmutató **5**

3.1. Fájllista . . . 5

4. Adatszerkezetek dokumentációja **7**

4.1. Ablak_info struktúrareferencia . . . 7

4.1.1. Részletes leírás . . . 7

4.2. Harom_hely struktúrareferencia . . . 7

4.2.1. Részletes leírás . . . 8

4.3. OszlopSor_Lista_Elem struktúrareferencia . . . 8

4.3.1. Részletes leírás . . . 8

4.4. Tabla struktúrareferencia . . . 9

4.4.1. Részletes leírás . . . 9

5. Fájlok dokumentációja **11**

5.1. GoL_graphics.c fájlreferencia . . . 11

5.1.1. Részletes leírás . . . 12

5.1.2. Függvények dokumentációja . . . 12

5.1.2.1. jatek() . . . 12

5.1.2.2. jatek_kattint() . . . 12

5.1.2.3. jatek_nextgen() . . . 14

5.1.2.4. menu() . . . 14

5.1.2.5. sdl_init() . . . 14

5.1.2.6. tabla_meret() . . . 15

5.1.2.7. xy_in_rect() . . . 15

5.2. GoL_graphics.h fájlreferencia . . . 15

5.2.1. Részletes leírás . . . 16

5.2.2. Enumerációk dokumentációja . . . 17

5.2.2.1. Allapot . . . 17

5.2.3. Függvények dokumentációja . . . 17

5.2.3.1. jatek() . . . 17

5.2.3.2. jatek_kattint() . . . 17

5.2.3.3. jatek_nextgen() . . . 18

5.2.3.4. menu() . . . 18

5.2.3.5. sdl_init() . . . 18

5.2.3.6. tabla_meret() . . . 19
5.2.3.7. xy_in_rect() . . . 19
5.3. GoL_graphics.h . . . 19
5.4. GoL_logics.c fájlreferencia . . . 20
5.4.1. Részletes leírás . . . 21
5.4.2. Függvények dokumentációja . . . 21
5.4.2.1. destroy_tabla() . . . 21
5.4.2.2. flip() . . . 21
5.4.2.3. init_tabla() . . . 22
5.4.2.4. uj_generacio() . . . 22
5.5. GoL_logics.h fájlreferencia . . . 23
5.5.1. Részletes leírás . . . 23
5.5.2. Függvények dokumentációja . . . 24
5.5.2.1. destroy_tabla() . . . 24
5.5.2.2. flip() . . . 25
5.5.2.3. init_tabla() . . . 25
5.5.2.4. uj_generacio() . . . 25
5.6. GoL_logics.h . . . 26
5.7. GoL_main.c fájlreferencia . . . 26
5.7.1. Részletes leírás . . . 27

Tárgymutató **29**

1. fejezet

Specifikacio

Nagy Ábel, CPD 63P
Prog1 NHF2, **Game of Life specifikáció**

1.1. Mi fogadja a felhasználót a program megnyitásakor?

- Grafikus UI, a felhasználó nem lát konzolt
- 3 opciós menü: játék, betöltés, súgó
- A játék menüpont előhoz egy grafikus felületet, ahol a kezdőállapotot lehet megadni
- A Betöltés menüpontban egy korábbi kezdőállapotot lehet megnyitni. (Alapértelmezett, vagy korábban felh. által létrehozott.)
- A súgó menüpont egy rövid szöveges leírást jelenít meg a GoL játékról.

1.1.1. 1. "Játék"

- A felhasználó mindenek előtt megadja a játéktér dimenzióit (szélesség, magasság). A felhasználó javaslatot kap a játéktér méretére, az ablak méretétől függően.
- Kívánt méretű négyzetháló jelenik meg. A cellákra kattintva a játékos váltogathatja, hogy a cellák élők (világos), vagy halottak (sötét) legyenek a szimuláció kezdetekor.
- Az ablak szélén egy mentés ikonnal lehetőség van elmenteni a játéktér jelenlegi állapotát. Ez a játéktér létrzése közben végig jelen van. Erre kattintáskor a felhasználó nevet kell adjon a mentésének.
- Az ablak szélén van egy lejátszás gomb. Ennek megnyomása után elindul a szimuláció futtatása. A gomb megnyomása után a gomb átváltozik megállít gombbá, ami megállítja a szimuláció futását. A futó szimulációban az állapotok rövid szünettel váltakoznak.
- Az ablak szélén a lejátszik/megállít gomb közelében van egy "következő állapot" gomb. Ez betölti a szimuláció következő állapotát.
- Az escape gombe megnyomásával vissza lehet térni a menübe.

1.1.2. 2. "Betöltés"

- Megnézi milyen fájlok vannak a ./saves mappában, ezekből felajánl egy listát, ahonnan a felhasználó kiválaszthat egy mentést, amit betölt. A listában a mentések neve jelenik meg.
- A listában megjelenik néhány alapértelmezett állapot, amelyek a Wikipédia oldal általi példák alapján készültek.
- A betöltött játék megnyitja a játéktért, ahol a cellák olyan állapotban vannak mint a mentés pillanatában voltak.
- Betöltés előtt a játék ellenőrzi a betöltendő mentés formai helyességét, hibaüzenetet dob ha az nem helyes.
- Az escape gombe megnyomásával vissza lehet térni a menübe.

1.1.3. 3. "Súgó"

- A súgó kiírja a GoL működésének szabályait és keletkezésének történetét, a Wikipédia oldal segítségével.
- Mutat egy képet John Conwayról, rest in peace üzenettel

1.2. A mentett játékállások

A mentett játékállapotok txt fájlok, a játék mappáján belüli ./saves mappában.
Az adatok az alábbi formátumban tárolódnak: (A 0-s jegyek az élettelen, az 1-esek az élő cellákat jelzik.)
{verziószám}
{szélesség} {magasság}
0 0 0 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 0 0 0

2. fejezet

Adatszerkezet-mutató

2.1. Adatszerkezetek

Az összes adatszerkezet listája rövid leírásokkal:

Ablak_info
: Az Ablakra vonatkozó minden alapvető tulajdonság 7

Harom_hely
: 3 SDL_Rect, amik a renderer-n belül megmondják hol vannak a gombok. Az s_menu állapotban használatos 7

OszlopSor_Lista_Elem
: Linked list elem oszloppal és sorral A megváltoztatott cellák listájának kezelésére 8

Tabla
: A GoL játéktábla tárolására való struct 9

3. fejezet

Fájlmutató

3.1. Fájllista

Az összes dokumentált fájl listája rövid leírásokkal:

GoL_graphics.c
A program grafikáját szabályozzó fájl. Felelős mindenért, ami a megjelenítés része 11
GoL_graphics.h
A program megjelenítéséhez használt funkciókat és structokat leíró headerfájl 15
GoL_logics.c
A játék fő logikáját leíró fájl. A játéktábla manipulálásával foglalkozik 20
GoL_logics.h
A játék logikájában használt függvényeket és structokat leíró header fájl 23
GoL_main.c
A program futtatásához szükséges main loop fájlaja 26

4. fejezet

Adatszerkezetek dokumentációja

4.1. Ablak_info struktúrareferencia

Az Ablakra vonatkozó minden alapvető tulajdonság.

`#include <GoL_graphics.h>`

Adatmezők

- SDL_Renderer ∗ **renderer**
- Allapot **state**
- int **width_screen**
- int **height_screen**

4.1.1. Részletes leírás

Az Ablakra vonatkozó minden alapvető tulajdonság.

Paraméterek

renderer	
state	
width_screen	
height_screen	

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- GoL_graphics.h

4.2. Harom_hely struktúrareferencia

3 SDL_Rect, amik a renderer-n belül megmondják hol vannak a gombok. Az s_menu állapotban használatos.

`#include <GoL_graphics.h>`

Adatmezők

- SDL_Rect **j**
- SDL_Rect **b**
- SDL_Rect **s**

4.2.1. Részletes leírás

3 SDL_Rect, amik a renderer-n belül megmondják hol vannak a gombok. Az s_menu állapotban használatos.

Paraméterek

j	A 'Játék' gomb jelenlegi helye a képernyőn
b	A 'Betölt' gomb jelenlegi helye a képernyőn
s	A 'Súgó' gomb jelenlegi helye a képernyőn

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- GoL_graphics.h

4.3. OszlopSor_Lista_Elem struktúrareferencia

Linked list elem oszloppal és sorral A megváltoztatott cellák listájának kezelésére.

`#include <GoL_logics.h>`

Adatmezők

- int **sor**
- int **oszlop**
- struct OszlopSor_Lista_Elem * **next**

4.3.1. Részletes leírás

Linked list elem oszloppal és sorral A megváltoztatott cellák listájának kezelésére.

Paraméterek

sor	
oszlop	
next	következő listaelemre mutató pointer

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- GoL_logics.h

4.4. Tabla struktúrareferencia

A GoL játéktábla tárolására való struct.

`#include <GoL_logics.h>`

Adatmezők

- int ** **g**
- int **sz**
- int **m**
- SDL_Rect ** **rects**

4.4.1. Részletes leírás

A GoL játéktábla tárolására való struct.

Paraméterek

g	grid (2d array), 0/1 (halott/élő)
sz	szélesség
m	magasság
rects	A képernyőn megjelenített cellák listája (2d array)

Ez a dokumentáció a struktúráról a következő fájl alapján készült:

- GoL_logics.h

5. fejezet

Fájlok dokumentációja

5.1. GoL_graphics.c fájlreferencia

A program grafikáját szabályozzó fájl. Felelős mindenért, ami a megjelenítés része.

```
#include <stdio.h>
#include <stdlib.h>
#include <SDL2/SDL.h>
#include <SDL2/SDL2_gfxPrimitives.h>
#include <SDL2/SDL_ttf.h>
#include <math.h>
#include "src/debugmalloc.h"
#include "GoL_logics.h"
#include "GoL_graphics.h"
```

Függvények

- void sdl_init (Ablak_info ∗env)

  Inicializálja az SDL-t, betölti az alapállapotokat az Ablak_info objektumba. Figyel a lehetséges hibákra, és hibaüzenettel kilép, ha fellépnek.
- int xy_in_rect (const int x, const int y, SDL_Rect rect)

  Megvizsgálja hogy az (x,y) koordináta (általában a kurzor) az SDL_Rect elemen belül található -e (returns 1 or 0)
- void menu (Ablak_info ∗env, TTF_Font ∗font_menu, Harom_hely ∗gombok_helye)

  Megváltoztatja az Ablak_info objektum state-jét s_menu-re. Eltüntet bármit ami épp a képernyőn van és kirajzolja a főmenüt.
- void tabla_meret (Ablak_info ∗env, Tabla ∗t)

  Megváltoztatja az Ablak_info objektum state-jét s_tabla_meret-re. Eltüntet bármit ami épp a képernyőn van és kirajzolja a tábla méretét kiválasztó felületet. Inicializálja a Tabla objektumot a megadott adatokkal.
- void jatek (Ablak_info ∗env, Tabla ∗t)

  Megváltoztatja az Ablak_info objektum state-jét s_jatek-ra. Eltüntet bármit ami épp a képernyőn van és kirajzolja a játéktáblát.
- void jatek_kattint (Ablak_info ∗env, Tabla ∗t, const int x, const int y)

  Ha az (x,y) koordináta egy cella belsejében van, megváltoztatja annak állapotát. A s_jatek állapotban használatos, kattintás ellenőrzésére.
- void jatek_nextgen (Ablak_info ∗env, Tabla ∗t)

  A szimulációt a következő állapotra lápteti. Ki is rajzolja a változásokat.

5.1.1. Részletes leírás

A program grafikáját szabályozzó fájl. Felelős mindenért, ami a megjelenítés része.

Szerző

Nagy Ábel (CPD63P) (nagy.abel@edu.bme.hu)

Verzió

0.2

Dátum

2021-11-10

Szerzői jog

Copyright (c) 2021

5.1.2. Függvények dokumentációja

5.1.2.1. jatek()

```
void jatek (
            Ablak_info * env,
            Tabla * t )
```

Megváltoztatja az Ablak_info objektum state-jét s_jatek-ra. Eltüntet bármit ami épp a képernyőn van és kirajzolja a játéktáblát.

Paraméterek

env	
t	

5.1.2.2. jatek_kattint()

```
void jatek_kattint (
            Ablak_info * env,
            Tabla * t,
            int x,
            int y )
```

Ha az (x,y) koordináta egy cella belsejében van, megváltoztatja annak állapotát. A s_jatek állapotban használatos, kattintás ellenőrzésére.

Paraméterek

env	
t	
x	
y	

5.1.2.3. jatek_nextgen()

```
void jatek_nextgen (
            Ablak_info * env,
            Tabla * t )
```

A szimulációt a következő állapotra lápteti. Ki is rajzolja a változásokat.

Paraméterek

env	
t	

5.1.2.4. menu()

```
void menu (
            Ablak_info * env,
            TTF_Font * font_menu,
            Harom_hely * gombok_helye )
```

Megváltoztatja az Ablak_info objektum state-jét s_menu-re. Eltüntet bármit ami épp a képernyőn van és kirajzolja a főmenüt.

Paraméterek

env	
font_menu	Egy betöltött betűtípus
gombok_helye	

5.1.2.5. sdl_init()

```
void sdl_init (
            Ablak_info * env )
```

Inicializálja az SDL-t, betölti az alapállapotokat az Ablak_info objektumba. Figyel a lehetséges hibákra, és hibaüzenettel kilép, ha fellépnek.

Paraméterek

env	

5.1.2.6. tabla_meret()

```
void tabla_meret (
            Ablak_info * env,
            Tabla * t )
```

Megváltoztatja az Ablak_info objektum state-jét s_tabla_meret-re. Eltüntet bármit ami épp a képernyőn van és kirajzolja a tábla méretét kiválasztó felületet. Inicializálja a Tabla objektumot a megadott adatokkal.

Paraméterek

env	
t	

5.1.2.7. xy_in_rect()

```
int xy_in_rect (
            const int x,
            const int y,
            SDL_Rect rect )
```

Megvizsgálja hogy az (x,y) koordináta (általában a kurzor) az SDL_Rect elemen belül található -e (returns 1 or 0)

Paraméterek

x	
y	
rect	

Visszatérési érték

1 or 0

5.2. GoL_graphics.h fájlreferencia

A program megjelenítéséhez használt funkciókat és structokat leíró headerfájl.

```
#include "GoL_logics.h"
#include <SDL2/SDL.h>
#include <SDL2/SDL2_gfxPrimitives.h>
#include <SDL2/SDL_ttf.h>
```

Adatszerkezetek

- struct Ablak_info

  Az Ablakra vonatkozó minden alapvető tulajdonság.
- struct Harom_hely

  3 SDL_Rect, amik a renderer-n belül megmondják hol vannak a gombok. Az s_menu állapotban használatos.

Enumerációk

- enum Allapot {
  s_menu , s_tabla_meret , s_jatek , s_betolt ,
  s_sugo }

  A program jelenlegi állapotát leíró enum.

Függvények

- void sdl_init (Ablak_info ∗env)

  Inicializálja az SDL-t, betölti az alapállapotokat az Ablak_info objektumba. Figyel a lehetséges hibákra, és hibaüzenettel kilép, ha fellépnek.
- int xy_in_rect (const int x, const int y, SDL_Rect rect)

  Megvizsgálja hogy az (x,y) koordináta (általában a kurzor) az SDL_Rect elemen belül található -e (returns 1 or 0)
- void menu (Ablak_info ∗env, TTF_Font ∗font_menu, Harom_hely ∗gombok_helye)

  Megváltoztatja az Ablak_info objektum state-jét s_menu-re. Eltüntet bármit ami épp a képernyőn van és kirajzolja a főmenüt.
- void tabla_meret (Ablak_info ∗env, Tabla ∗t)

  Megváltoztatja az Ablak_info objektum state-jét s_tabla_meret-re. Eltüntet bármit ami épp a képernyőn van és kirajzolja a tábla méretét kiválasztó felületet. Inicializálja a Tabla objektumot a megadott adatokkal.
- void jatek (Ablak_info ∗env, Tabla ∗t)

  Megváltoztatja az Ablak_info objektum state-jét s_jatek-ra. Eltüntet bármit ami épp a képernyőn van és kirajzolja a játéktáblát.
- void jatek_kattint (Ablak_info ∗env, Tabla ∗t, int x, int y)

  Ha az (x,y) koordináta egy cella belsejében van, megváltoztatja annak állapotát. A s_jatek állapotban használatos, kattintás ellenőrzésére.
- void jatek_nextgen (Ablak_info ∗env, Tabla ∗t)

  A szimulációt a következő állapotra lápteti. Ki is rajzolja a változásokat.

5.2.1. Részletes leírás

A program megjelenítéséhez használt funkciókat és structokat leíró headerfájl.

Szerző

Nagy Ábel (CPD63P) (nagy.abel@edu.bme.hu)

Verzió

0.1

Dátum

2021-11-10

Szerzői jog

5.2.2. Enumerációk dokumentációja

5.2.2.1. Allapot

```
enum Allapot
```

A program jelenlegi állapotát leíró enum.

Enumeráció-értékek

s_menu	A menü a három gombbal
s_tabla_meret	A tábla méretének megadása a játék gombra kattintás után
s_jatek	A játéktábla
s_betolt	A mentés választó menü, a betöltés gomb megnyomása után
s_sugo	A súgó, a súgó gomb megnyomása után

5.2.3. Függvények dokumentációja

5.2.3.1. jatek()

```
void jatek (
            Ablak_info * env,
            Tabla * t )
```

Megváltoztatja az Ablak_info objektum state-jét s_jatek-ra. Eltüntet bármit ami épp a képernyőn van és kirajzolja a játéktáblát.

Paraméterek

env	
t	

5.2.3.2. jatek_kattint()

```
void jatek_kattint (
            Ablak_info * env,
            Tabla * t,
            int x,
            int y )
```

Ha az (x,y) koordináta egy cella belsejében van, megváltoztatja annak állapotát. A s_jatek állapotban használatos, kattintás ellenőrzésére.

Paraméterek

env	
t	
x	
y	

5.2.3.3. jatek_nextgen()

```
void jatek_nextgen (
            Ablak_info * env,
            Tabla * t )
```

A szimulációt a következő állapotra lápteti. Ki is rajzolja a változásokat.

Paraméterek

env	
t	

5.2.3.4. menu()

```
void menu (
            Ablak_info * env,
            TTF_Font * font_menu,
            Harom_hely * gombok_helye )
```

Megváltoztatja az Ablak_info objektum state-jét s_menu-re. Eltüntet bármit ami épp a képernyőn van és kirajzolja a főmenüt.

Paraméterek

env	
font_menu	Egy betöltött betűtípus
gombok_helye	

5.2.3.5. sdl_init()

```
void sdl_init (
            Ablak_info * env )
```

Inicializálja az SDL-t, betölti az alapállapotokat az Ablak_info objektumba. Figyel a lehetséges hibákra, és hibaüzenettel kilép, ha fellépnek.

Paraméterek

env	

5.2.3.6. tabla_meret()

```
void tabla_meret (
            Ablak_info * env,
            Tabla * t )
```

Megváltoztatja az Ablak_info objektum state-jét s_tabla_meret-re. Eltüntet bármit ami épp a képernyőn van és kirajzolja a tábla méretét kiválasztó felületet. Inicializálja a Tabla objektumot a megadott adatokkal.

Paraméterek

env	
t	

5.2.3.7. xy_in_rect()

```
int xy_in_rect (
            const int x,
            const int y,
            SDL_Rect rect )
```

Megvizsgálja hogy az (x,y) koordináta (általában a kurzor) az SDL_Rect elemen belül található -e (returns 1 or 0)

Paraméterek

x	
y	
rect	

Visszatérési érték

1 or 0

5.3. GoL_graphics.h

Ugrás a fájl dokumentációjához.

```
1
12 #include "GoL_logics.h"
13 #include <SDL2/SDL.h>
14 #include <SDL2/SDL2_gfxPrimitives.h>
15 #include <SDL2/SDL_ttf.h>
```

```
16
17 #ifndef GOL_GRAPHICS_H
18 #define GOL_GRAPHICS_H
19
23 typedef enum Allapot{
25     s_menu,
27     s_tabla_meret,
29     s_jatek,
31     s_betolt,
33     s_sugo
34 }Allapot;
35
44 typedef struct Ablak_info{
45     SDL_Renderer *renderer;
46     Allapot state;
47     int width_screen, height_screen;
48 }Ablak_info;
49
57 typedef struct Harom_hely{
58     SDL_Rect j, b, s;
59 }Harom_hely;
60
66 void sdl_init(Ablak_info *env);
75 int xy_in_rect(const int x, const int y, SDL_Rect rect);
83 void menu(Ablak_info *env, TTF_Font *font_menu, Harom_hely *gombok_helye);
91 void tabla_meret(Ablak_info *env, Tabla *t);
98 void jatek(Ablak_info *env, Tabla *t);
107 void jatek_kattint(Ablak_info *env, Tabla *t, int x, int y);
114 void jatek_nextgen(Ablak_info *env, Tabla *t);
115
116 #endif
```

5.4. GoL_logics.c fájlreferencia

A játék fő logikáját leíró fájl. A játéktábla manipulálásával foglalkozik.

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <SDL2/SDL.h>
#include "GoL_logics.h"
#include "src/debugmalloc.h"
```

Függvények

- void init_tabla (Tabla *ujtabla, int szelesseg, int magassag)

  Létrehozza, feltölti nullákkal a Tabla objektumot A tábla sz és m paraméterei nem egyeznek a magassággal, hiszen extra nullásokból álló "szegélyt" kap köré
- void destroy_tabla (Tabla *regitabla)

  Felszabadítja a Tabla objektum memóriahelyét.
- void flip (Tabla *t, int sor, int oszlop)

  A megadott sorban és oszlopban lévő cella értékét megváltoztatja (Halott -> Élő, Élő -> Halott)
- OszlopSor_Lista_Elem * uj_generacio (Tabla *t)

  A szimulációt továbblépteti a következő állásba. A szegély cellái nem változnak, ezek tulajdonképpen nem a szimuláció részei. Linked list (első elemét) ad vissza, a megváltoztatott cellák koordinátájával.

5.4.1. Részletes leírás

A játék fő logikáját leíró fájl. A játéktábla manipulálásával foglalkozik.

Szerző

Nagy Ábel (CPD63P) (nagy.abel@edu.bme.hu)

Verzió

0.1

Dátum

2021-11-08

Szerzői jog

5.4.2. Függvények dokumentációja

5.4.2.1. destroy_tabla()

```
void destroy_tabla (
            Tabla * regitabla )
```

Felszabadítja a Tabla objektum memóriahelyét.

Paraméterek

regitabla	

5.4.2.2. flip()

```
void flip (
            Tabla * t,
            int sor,
            int oszlop )
```

A megadott sorban és oszlopban lévő cella értékét megváltoztatja (Halott -> Élő, Élő -> Halott)

Paraméterek

t	
sor	
oszlop	

5.4.2.3. init_tabla()

```
void init_tabla (
            Tabla * ujtabla,
            int szelesseg,
            int magassag )
```

Létrehozza, feltölti nullákkal a Tabla objektumot A tábla sz és m paraméterei nem egyeznek a magassággal, hiszen extra nullásokból álló "szegélyt" kap köré

Figyelmeztetés

Használat után törlendő memóriaszemetet hagy!!

Paraméterek

ujtabla	
szelesseg	
magassag	

5.4.2.4. uj_generacio()

```
OszlopSor_Lista_Elem * uj_generacio (
            Tabla * t )
```

A szimulációt továbblépteti a következő állásba. A szegély cellái nem változnak, ezek tulajdonképpen nem a szimuláció részei. Linked list (első elemét) ad vissza, a megváltoztatott cellák koordinátájával.

Figyelmeztetés

Felszabadítandó memóriaszemetet hagy (a linked listtel)!!

Paraméterek

t	

Visszatérési érték

a megváltoztatott cellák

5.5. GoL_logics.h fájlreferencia

A játék logikájában használt függvényeket és structokat leíró header fájl.

Adatszerkezetek

- struct Tabla

  A GoL játéktábla tárolására való struct.
- struct OszlopSor_Lista_Elem

  Linked list elem oszloppal és sorral A megváltoztatott cellák listájának kezelésére.

Függvények

- void init_tabla (Tabla ∗ujtabla, int szelesseg, int magassag)

  Létrehozza, feltölti nullákkal a Tabla objektumot A tábla sz és m paraméterei nem egyeznek a magassággal, hiszen extra nullásokból álló "szegélyt" kap köré
- void destroy_tabla (Tabla ∗regitabla)

  Felszabadítja a Tabla objektum memóriahelyét.
- void flip (Tabla ∗t, int sor, int oszlop)

  A megadott sorban és oszlopban lévő cella értékét megváltoztatja (Halott -> Élő, Élő -> Halott)
- OszlopSor_Lista_Elem ∗ uj_generacio (Tabla ∗t)

  A szimulációt továbblépteti a következő állásba. A szegély cellái nem változnak, ezek tulajdonképpen nem a szimuláció részei. Linked list (első elemét) ad vissza, a megváltoztatott cellák koordinátájával.

5.5.1. Részletes leírás

A játék logikájában használt függvényeket és structokat leíró header fájl.

Szerző

Nagy Ábel (CPD63P) (nagy.abel@edu.bme.hu)

Verzió

0.1

Dátum

2021-11-08

Szerzői jog

Copyright (c) 2021

5.5.2. Függvények dokumentációja

5.5.2.1. destroy_tabla()

```
void destroy_tabla (
            Tabla * regitabla )
```

Felszabadítja a Tabla objektum memóriahelyét.

Paraméterek

regitabla	

5.5.2.2. flip()

```
void flip (
            Tabla * t,
            int sor,
            int oszlop )
```

A megadott sorban és oszlopban lévő cella értékét megváltoztatja (Halott -> Élő, Élő -> Halott)

Paraméterek

t	
sor	
oszlop	

5.5.2.3. init_tabla()

```
void init_tabla (
            Tabla * ujtabla,
            int szelesseg,
            int magassag )
```

Létrehozza, feltölti nullákkal a Tabla objektumot A tábla sz és m paraméterei nem egyeznek a magassággal, hiszen extra nullásokból álló "szegélyt" kap köré

Figyelmeztetés

Használat után törlendő memóriaszemetet hagy!!

Paraméterek

ujtabla	
szelesseg	
magassag	

5.5.2.4. uj_generacio()

```
OszlopSor_Lista_Elem * uj_generacio (
```

```
            Tabla * t )
```

A szimulációt továbblépteti a következő állásba. A szegély cellái nem változnak, ezek tulajdonképpen nem a szimuláció részei. Linked list (első elemét) ad vissza, a megváltoztatott cellák koordinátájával.

Figyelmeztetés

Felszabadítandó memóriaszemetet hagy (a linked listtel)!!

Paraméterek

t	

Visszatérési érték

a megváltoztatott cellák

5.6. GoL_logics.h

Ugrás a fájl dokumentációjához.

```
1
12 #ifndef GOL_LOGICS_H
13 #define GOL_LOGICS_H
14
23 typedef struct Tabla{
24     int** g;
25     int sz, m;
26     SDL_Rect** rects;
27 }Tabla;
28
36 typedef struct OszlopSor_Lista_Elem{
37     int sor, oszlop;
38     struct OszlopSor_Lista_Elem* next;
39 }OszlopSor_Lista_Elem;
40
49 void init_tabla(Tabla* ujtabla, int szelesseg, int magassag);
50
55 void destroy_tabla(Tabla* regitabla);
56
63 void flip(Tabla* t, int sor, int oszlop);
64
73 OszlopSor_Lista_Elem* uj_generacio(Tabla* t);
74
75 #endif
```

5.7. GoL_main.c fájlreferencia

A program futtatásához szükséges main loop fájlaja.

```
#include <stdio.h>
#include <stdlib.h>
#include <SDL2/SDL.h>
#include <SDL2/SDL2_gfxPrimitives.h>
#include <SDL2/SDL_ttf.h>
#include <math.h>
#include "src/debugmalloc.h"
#include "GoL_logics.h"
#include "GoL_graphics.h"
```

Függvények

- int **main** (void)

5.7.1. Részletes leírás

A program futtatásához szükséges main loop fájlaja.

Szerző

Nagy Ábel (CPD63P) (nagy.abel@edu.bme.hu)

Verzió

0.1

Dátum

2021-11-10

Szerzői jog

Tárgymutató

Ablak_info, 7
Allapot
    GoL_graphics.h, 17

destroy_tabla
    GoL_logics.c, 21
    GoL_logics.h, 24

flip
    GoL_logics.c, 21
    GoL_logics.h, 25

GoL_graphics.c, 11
    jatek, 12
    jatek_kattint, 12
    jatek_nextgen, 14
    menu, 14
    sdl_init, 14
    tabla_meret, 15
    xy_in_rect, 15
GoL_graphics.h, 15
    Allapot, 17
    jatek, 17
    jatek_kattint, 17
    jatek_nextgen, 18
    menu, 18
    s_betolt, 17
    s_jatek, 17
    s_menu, 17
    s_sugo, 17
    s_tabla_meret, 17
    sdl_init, 18
    tabla_meret, 19
    xy_in_rect, 19
GoL_logics.c, 20
    destroy_tabla, 21
    flip, 21
    init_tabla, 22
    uj_generacio, 22
GoL_logics.h, 23
    destroy_tabla, 24
    flip, 25
    init_tabla, 25
    uj_generacio, 25
GoL_main.c, 26

Harom_hely, 7

init_tabla
    GoL_logics.c, 22
    GoL_logics.h, 25

jatek
    GoL_graphics.c, 12
    GoL_graphics.h, 17
jatek_kattint
    GoL_graphics.c, 12
    GoL_graphics.h, 17
jatek_nextgen
    GoL_graphics.c, 14
    GoL_graphics.h, 18

menu
    GoL_graphics.c, 14
    GoL_graphics.h, 18

OszlopSor_Lista_Elem, 8

s_betolt
    GoL_graphics.h, 17
s_jatek
    GoL_graphics.h, 17
s_menu
    GoL_graphics.h, 17
s_sugo
    GoL_graphics.h, 17
s_tabla_meret
    GoL_graphics.h, 17
sdl_init
    GoL_graphics.c, 14
    GoL_graphics.h, 18

Tabla, 9
tabla_meret
    GoL_graphics.c, 15
    GoL_graphics.h, 19

uj_generacio
    GoL_logics.c, 22
    GoL_logics.h, 25

xy_in_rect
    GoL_graphics.c, 15
    GoL_graphics.h, 19