

# Programozói dokumentáció

## A *graphics* modul leírása

---

Ennek a modulnak a legfőbb feladata a grafikus interfész megvalósítása, megjelenítése az SDL grafikai funkciók segítségével. Az *integerek* használatán kívül az *SDL surface* használata a lényeges, amely az egyes SDL grafikai függvények pointere a képernyőre.

Függvények:

`void LIVEBOX(int x, int y);` és `void DEADBOX(int x, int y);`

Ez a két függvény az egyes cellákat rakja be, két hívója a *PRINTCELLS* (*base* modul) és a *Drawer* (*general* modul). két integer koordináta alapján helyezi el a boxokat, pixelszámolással. A cellák színét a *theme* változó alapján állítja be egy *switch*-csel aminek értékét a *Themeselect* függvénytől lekérdezve kapja meg.

`void MENULOAD();`

A felhasználói interfész, passzív gombjait helyezi el. Ez a menürendszer ahol a felhasználó meghívhatja grafikusan az egyes funkciókat.

`int Speed_slider(int x);`

Egy sebesség szabályozó, ami a koordinátái alapján változtatja a várakozási időt növelve vagy csökkentve azt. Koordinátáit az SDL egér koordinátái alapján kapja.

`void Run_Button(int *run);`

Egy egyszerű flip-flop ami elindítja vagy megállítja a *main*-ben lévő *Loop* (*general*) függvényt.

`int Themeselect(int x, int y);`

A megnyomott gomb alapján állítja be a statikus *theme* változóját, hogy azt majd a *LIVEBOX* és a *DEADBOX* függvények lekérdezzék.

## A *base* modul leírása

---

A *base* modul tartalmazza a fő algoritmusokat; A tömblefoglaló egységet, a matematikai számoló egységet, a kirajzoló funkciót és a léptető funkciót.

Adattípusok:

```
typedef struct Cell{  
    int state;  
    int rank;  
}Cell;
```

Ez egy cella típusa. A program elején a *SETUP* egy ilyen típusú 2 dimenziós tömböt foglal le. Ennek az adattípusnak az előnye hogy külön választhatjuk a sejt tulajdonságait megkönnyítve a számolásokat. A *rank* változó menti el a cella körülötte élő cellák számát, a *state* pedig alapjában véve egy logikai egység, aminek élő vagy halott (1/0) státusza lehet.

Főbb használói: *PRINTCELLS* –csupán kiolvas

*Rank\_and\_State\_calc* – számol vele, ír/olvas  
*SETUP*

```
typedef struct Coor{  
    int x;  
    int y;  
}Coo;
```

Egyszerű koordináta vektor. A program során megkönnyíti a változók átadását.

Mivel x és y változók sok helyen együtt vannak átadva ezért előnyös.

Legfőbb használata a koordináta alapú pálya-mozgatás.

Függvények:

```
void SETUP(Cell ***Table);
```

A konstansnak definiált *MAG* és *SZEL* (magasság-szélesség) alapján 1000\*1000-es 2 dimenziós tömböt foglal le dinamikusan *Cell* adattípusból.

Ez a tömb a játék alappillére. A függvény egyszer hívódik meg a játék elején.

A foglalást addig végzi, amíg nem kap érvényes pointert a tömbre.

void Rank\_and\_State\_calc(Cell \*\*Table);

A legfontosabb függvény. A függvény a tömbre mutató pointert kapja meg.

Az algoritmus a következő;

Körbejárja *for* ciklussal az egész tömböt kiszámolva az összes cellának a

*rankját* úgy hogy az értéke egyenlő a 8 szomszéd közül az élők számával.

Ezután a kapott érték és a játék szabály alapján meghatározza a cellák életét

ismét körbejárva a tömböt; Ha 3 a *rank* akkor biztosan élő, ha 2 akkor az

előző státusza marad. A többi esetben biztos, hogy 0 az értéke. Közben

lenullázza a *rank* ot hogy a következő számolásban helyes értékeket kapjunk.

void PRINTCELLS(Cell \*\*Table, Coor coo);

Az aktuális Coor koordináták alapján(amik a pálya pozícióját jelzik) körbejárja a

tömb megfelelő résztömbjét és a *LIVEBOX* vagy *DEADBOX* függvényt

meghívva grafikusán megjeleníti a tömböt.

void LOOP(Cell \*\*Table,Coor coo);

Meghívása egyenlő egy léptetéssel. Sorrendben a következő függvények

hívódnak;

Rank\_and\_State\_calc

Statistic

PRINTCELLS

## A *general* modul leírása

---

Ez a modul tartalmazza az egyéb funkciókat amiket leginkább a felhasználó hív meg, azaz aktív hívású funkciók. Ide tartozik a menürendszer működtetése és azok a függvények amelyek segítségével a felhasználó megváltoztathatja a játék fő tömbjének az adatait.

Ez a modul használja az előző két modult azaz ennek egyedüli definiálása automatikusan definiálja a fenti kettőt.

Függvények:

```
void Statistic(int n, Cell **Table);
```

A program összeszámolja az élő cellák számát és egy statikus változóban a generáció számot majd megjeleníti ezeket.

```
void NAV_BOX(Coo coo);
```

Egy segítő funkció amely megmutatja nekünk az aktuális helyzetünket a tömbben. Egyfajta térkép, tájékozódási szempontból előnyös.

```
void Clear(Cell **Table);
```

```
void Rand(Cell **Table);
```

Ez a két függvény működését tekintve nagyon hasonló bár funkciójukat tekintve külön választandóak. A *Clear* egyszerűen lenullázza a tömb *state* változóját, ezzel **MEGÖLVE az összes sejtet**

A *Rand* pedig véletlenszerűen feléleszt sejteket.

```
void Left_Click_Switch(int x, int y, int *run, Cell **Table, Coo coo);
```

Ez a függvény tartja fenn gombok működését; az egér koordinátái alapján kiszámolja hogy hova nyomotta felhasználó, majd meghívja a megfelelő funkciót, a megfelelő paraméterekkel. – tekinthető egy elosztónak.

```
void Draw(int x, int y, Cell **Table, Coo coo);
```

```
int Draw_state(int i);
```

A fenti két függvény szorosan összefügg. A Drawstate egy lekérdezhető flip-flop aminek állítgatása a rajzolás vagy törlés.