

DARÁZS-MODELL

Szlávi Péter

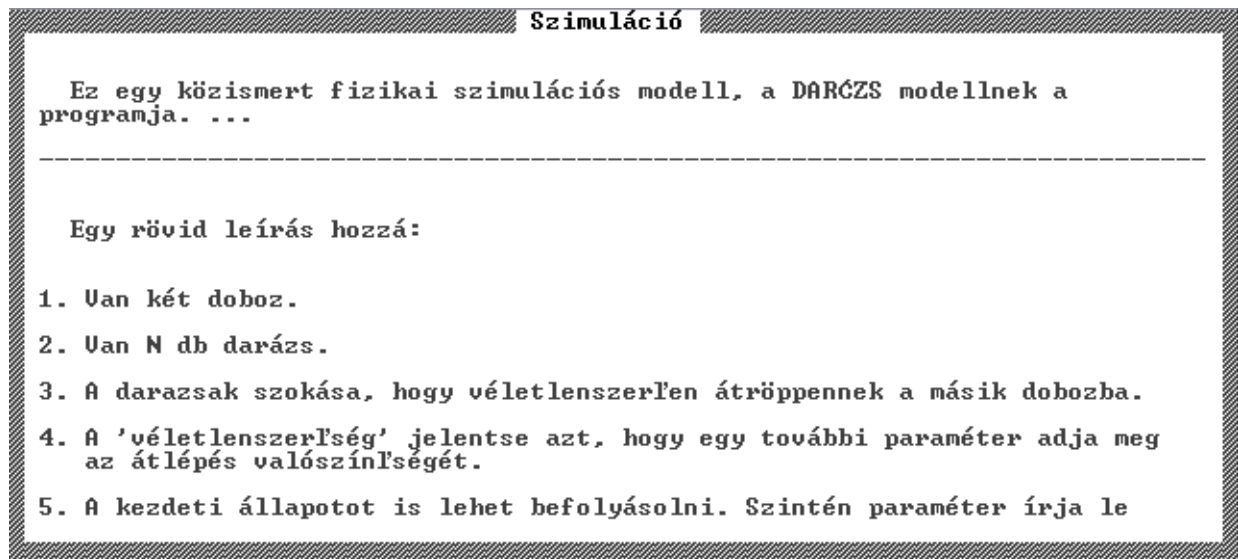
2000-2012

TARTALOM

DARÁZS-MODELL	1
1. Elvárások	2
2. Problémák – megoldásötletek	3
2.1. Állapotok, állapotváltoztatás, programkezelés	3
2.2. Képernyők	3
2.3. Billentyűkezelés	3
2.4. Tájékoztató	3
2.5. Grafikus képernyő	4
2.6. Összefoglalás	5
2.7. Szimulációs keret	5
3. A keretprogram	6
4. Feladatok	7
4.1. Alapfeladat	7
4.2. A jövőbe mutató célfeladat	7
4.3. Szorgalmi feladat	7
5. A teljes anyag	9

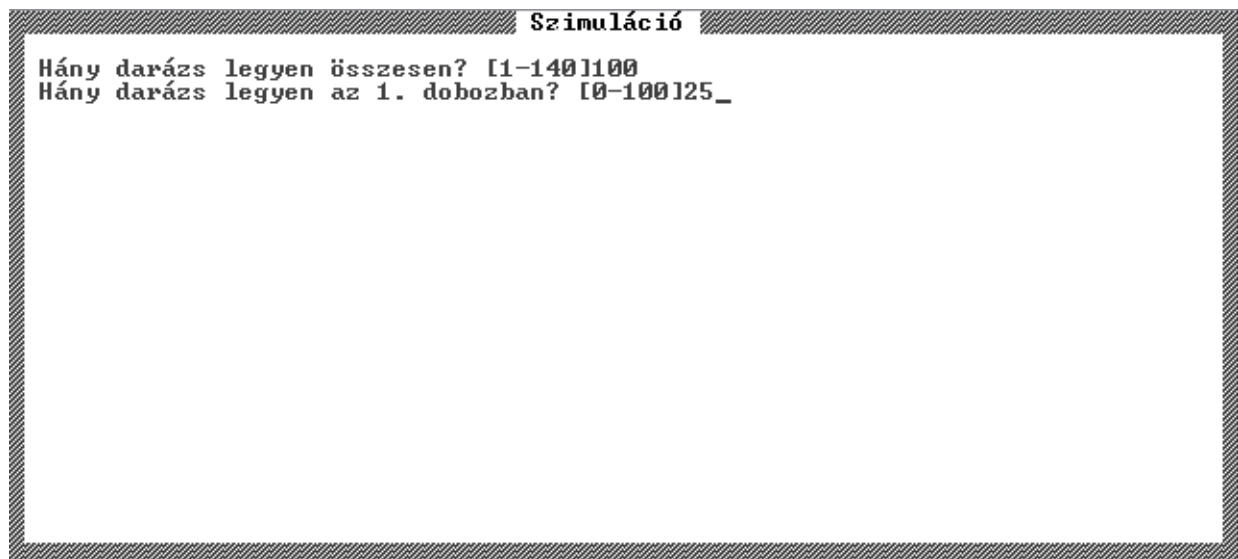
1. Elvárások

Az elvárásokat legjobban az alábbi, futás során keletkezett ábrásor fejezi ki.

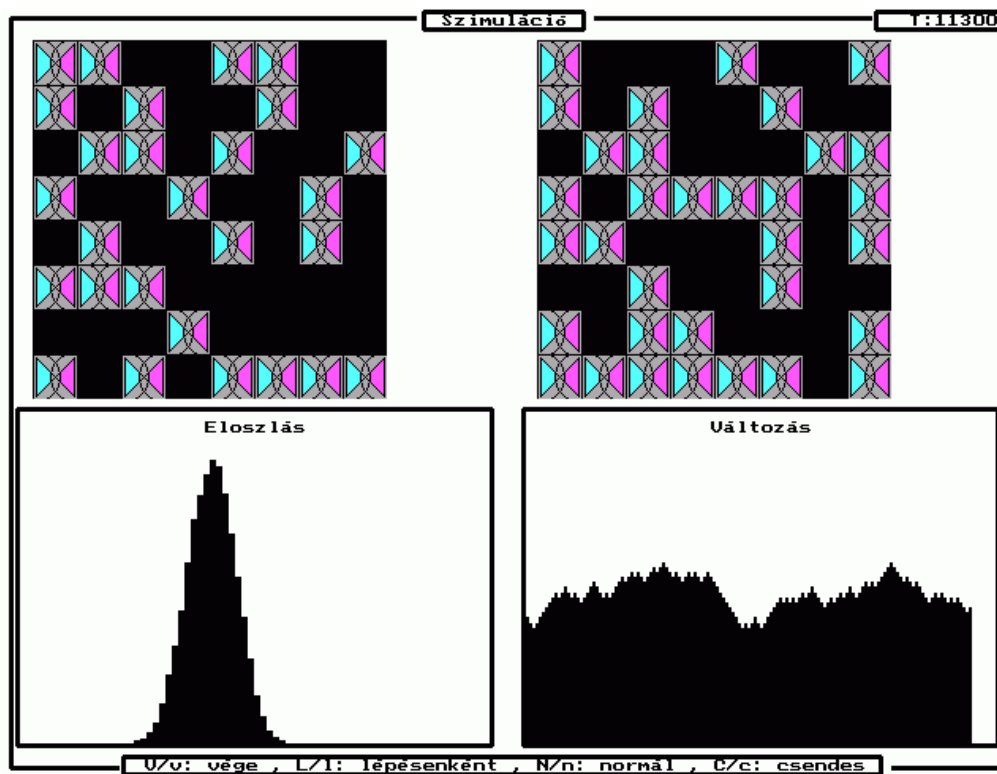


Lapozhatunk?/

1. ábra. Egy futási kép – Tájékoztató



2. ábra. Egy futási kép – Paraméterezés



3. ábra. Egy futási kép – Futás közben

Mindenekelőtt lássunk egy próbát! ([DarázsMo.exe](#))

2. Problémák – megoldásötletek

A kezelés és a látvány (a felület), valamint a szimuláció (a lényeg) specifikus problémái és megoldásötletei.

2.1. Állapotok, állapotváltoztatás, programkezelés

Normál / Lépésenkénti / Csendes / Vége

TűzemMódok típus ([felsorolás típus](#)), ÜzemMód állapotváltozó (TűzemMódok típusú)

2.2. Képernyők

Karakteres – tájékoztatás, paraméterezés

Grafikus – modellezés

Miért célszerű ez így? Miért nem csak karakteres vagy csak grafikus képernyő?

2.3. Billentyűkezelés

Billentyűfigyelés, várakozás-kijelzés, állapotmódosítás (ÜzemMód állapotváltozó)

2.4. Tájékoztató

Egyszerű szerkezetű (=előre a képernyőre tördelt) Text fájlból, karakteres képernyőre, lapozottan (de hátra, ill. előre több, mint egy lappal lépdelés nélkül).

Text-kezelési minimum – emlékeztető:

- Var f:Text; {karakterfájl} sor:String[80]; {egy sorát ebbe olvassuk majd be}
- Assign(f,fN); {\$i-} Reset(f); {\$i+} hiba:=IOResult<>0; {ellenőrzött nyitás}
- Eof(f) {=vége-e az f fájlnek?}
- Readln(f,sor); {egy sor beolvasása f-ből sor-ba}
- Close(f); {f lezárása}

Kicsit részletesebb összefoglaló itt található: [FajlosGyak.pdf](#).

2.5. Grafikus képernyő

Felépítése, állandó részei (keret, cím, idő, help-sor), ablakfajták (grafikon / eseménytér). L. a [3. ábrán!](#)

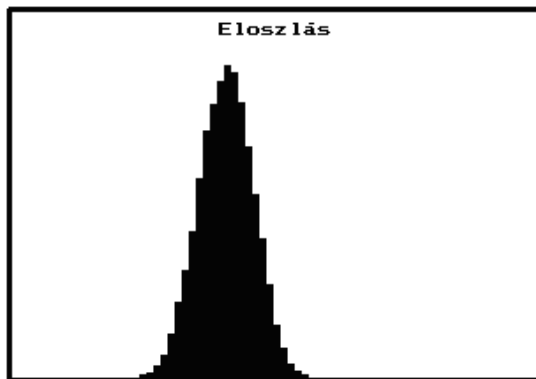
„Turbo” grafikai¹ „egyszer-egy” – emlékeztető:

- | | | |
|---------------------------------|-----------------|----------------------------|
| • Graph unit | • InitGraph | • Detect |
| • SetViewPort, GetMaxX... | • ClearViewPort | • SetLineStyle, SolidLn... |
| • SetTextJustify, CenterText... | • Rectangle | • OutTextXY |

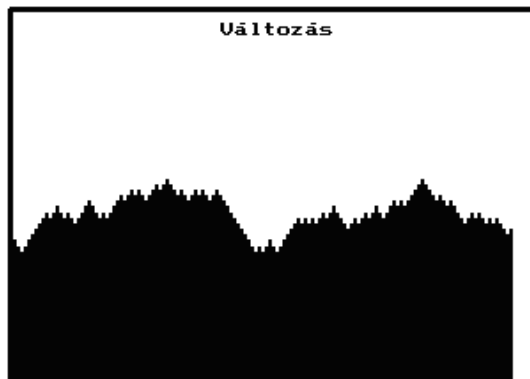
Részletesebb összefoglaló: [Graph unit](#).

A **Grafikon-ablak** műveletei:

- Létrehozás paraméterek szerint (hol, mekkora, cím, ...)
- (Első) kirajzolás (az ablak „statikus” részeinek megjelenítése)
- Teljes grafikon kirajzolás (pl. Csendes üzemmód után, vagy a teljes eloszlásdiagram)
- Eloszlásdiagram – csak a változó oszlop adminisztrálása, és üzemmódfüggő kirajzolása
- Idődiagram kirajzolás – az új oszloppal bővítés (adminisztrálással), és –üzemmódfüggő– kirajzolása shift-eléssel



4. ábra. Eloszlásdiagram



5. ábra. Idődiagram

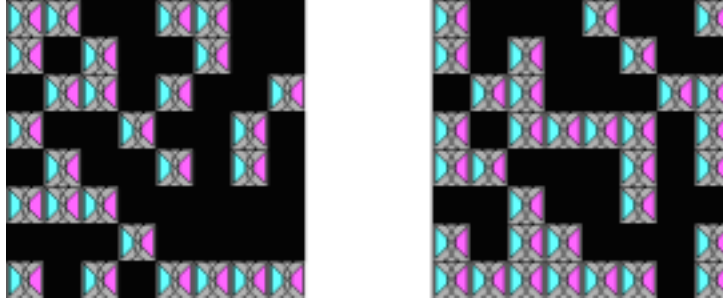
¹ A Free Pascal implementálói elfogadták a Borland Turbo grafikáját (TG), és egy az egyben megvalósították a saját rendszerükben.

TG részletesebb leírását itt találja: <http://pascal-programming.info/lesson8.php>

Darázs-modell

Az **Eseménytér-ablak** műveletei

- Eseménytér kirajzolása
- Egy elemi esemény követése



6. ábra. Eseménytér-ablak

2.6. Összefoglalás

A probléma forrása az, hogy bár előre tudhatók azok a végkifejletek, amelyek bekövetkezhetnek, de hogy mikor, és milyen mértékben azt a futás végén lehet csak biztosan tudni. Tehát a megjelenítendő információnak van **előre megfogalmazható szöveges** része, amelyekbe beleillesztendők a **futás aktualitásai**.

Megoldásötlet: a szövegben elhelyezni olyan **speciális jeleket**, amelyekről beolvasás közben „ordít”, hogy milyen aktualitást kell a helyére illeszteni. Tehát a megoldás algoritmus a hasonlít a tájékoztatóéra, azzal a nem kis különbséggel, hogy **soronként ellenőrizni** kell a speciális jelek meglétét, és esetleg **be kell oda illeszteni** a hozzátartozó **aktualitás** „értékét”.

2.7. Szimulációs keret

Modell-reprezentáció – a modell alapadatai:

- T – idő
- N, K – össz-darázsszám, bal dobozbeli darázsszám (*modell-specifikus*)
- UzemMod:TUzemModok – aktuális üzemmód
- Darazs – darazsak holléte (*modell-specifikus*)
- dobozTerkep – szabad/foglalt helyek nyilvántartása (*modell-specifikus*)
- darazsKep – egy darázs rajza (*modell-specifikus*)

Modell-implemetáció – a modell nagyvonalú algoritmus („pascalul”):

```
UzemMod:=Lepesenkenti;  
Tajekoztato;  
Inicializalas; T:=0;  
KezdoKepernyo;  
EredmenyMegjelenites;  
While not VegeE do  
  Begin  
    SzimulaciosLepes;  
    Inc(T);  
    EredmenyMegjelenites;  
  End;  
OsszefoglaloMegjelenites;
```

Az **eljárások/függvények** rövid leírása:

- **Tájékoztató:**
L. [feljebb](#)

- **Inicializálás:**
 - Paraméterbeolvasás (*modell-specifikus*)
 - Grafikonok parametrizálása
 - Eseménytér inicializálása
 - Darázkép-generálás (*modell-specifikus*)
 - Darazsak elhelyezése a modell-objektumokban (*modell-specifikus*)
- **KezdőKépernyő**
 - Grafikus képernyőre váltás
 - Grafikus képernyő (-alap)elkészítése (ablakok kirajzolása)
- **EredményMegjelenítés**
 - Aktuális idő kiírása
 - Grafikus ablakok frissítése (a változó adatok módosítása)
- **Vége**
 - Billentyűfigyelés –üzemmódtól függően: várakozás–
 - Végállapotfigyelés (*modell-specifikus*)
 - (Esetleg) kijelzés frissítés (Csendes → Normál / Lépésenkénti)
- **SzimulációsLépés**
 - A lényeg – modellváltozók módosítása (*modell-specifikus*)
 - Az eseménytér módosítása (ha kell)
- **ÖsszefoglalóMegjelenítés**
 - Konklúzió levonása – a végidőt, az össz- és a baloldali dobozbeli darázsszámot tartalmazó, előre gyártott, fájlban levő szöveg információkkal bővített megjelenítése (*a szöveg: modell-specifikus*)
 - L. a keretprogram OsszefoglaloMegjelenites eljárásában
 - Összefoglaló ábrák megjelenítése (*modell-specifikus*)

3. A keretprogram

L. [DarazsKe.pas](#).

Megértendő fogalmak, részek:

- a feltételes fordítás lényege és itteni célja
- a típusok – főleg a `THisztAblak`
- a globális adatok – főleg a `T`, `N`, `K`, `Darazs`, `DobozTerkep`
- az ablak-műveletek célja, paraméterezése
- az inicializálás feladatai (és a fejlesztési kényszer miatt keletkezett TP2FP... konverziós függvények)
- a 'kezdő képernyő' feladatai
- az eredménymegjelenítés feladatai
- a vége-e függvény (többlatszolgáltatásai)
- az 'összefoglaló megjelenítés' kulisszatitkai

4. Feladatok

4.1. Alapfeladat

A keretprogramban az `'(*ide kell a kód*)'` megjegyzéssel megjelölt részek kitöltése:

- `Tajekoztato` teljes megvalósítása
- `Inicializalas`-ban
 - az `N` és a `K` tisztességes beolvasása,
 - az eseménytér inicializálásából a darázsok elhelyezése a `Darazs`-ban és a `DobozTerkep` kitöltése
- `SzimulaciosLepes`-ben
 - az átrepülő darázs kiválasztása,
 - az eseménytér régi dobozából törlése,
 - az eseménytér új dobozában hely keresése, és elhelyezése

4.2. A jövőbe mutató célfeladat

A Darázs-modell keretprogramjából olyan **szimulációs keretprogram** elkészítése, amely a későbbiekben jobb kiindulópontul szolgálhatna, mint a Darázs-modellé. Azaz a biztosan fölösleges részeket ki kell hagyni. Annyit lehet előre tudni, hogy a későbbi modelljeinkben a „**mozgó alkatrészeket**” (a mostani darázsokat) már elegendő lesz **egy** valamilyen színes **pöttyel** jelölni, és csak egyetlen **reakcióterben** zajlanak majd az események (nem lesz több doboz).

4.3. Szorgalmi feladat

Illesszünk be egy **új diagramot** a programba, amelyen megjelenítjük azt az **eloszlást**, amelyet a darázs-modell a konkrét paraméterekhez jósol. Ezt elegendő egyetlen egyszer kirajzolni, hiszen az **idealizált végállapotot** szemlélteti. Az **összefoglalóban** kiírjuk, hogy kezdetben és a végén mennyi volt az **empirikus és az elméleti eloszlás „távolsága”**. (Az elméleti eloszlás $(N, \frac{1}{2})$ -paraméterű binomiális eloszlás.²)

Ezt a távolságot pl. az alábbi módon definiálhatjuk:

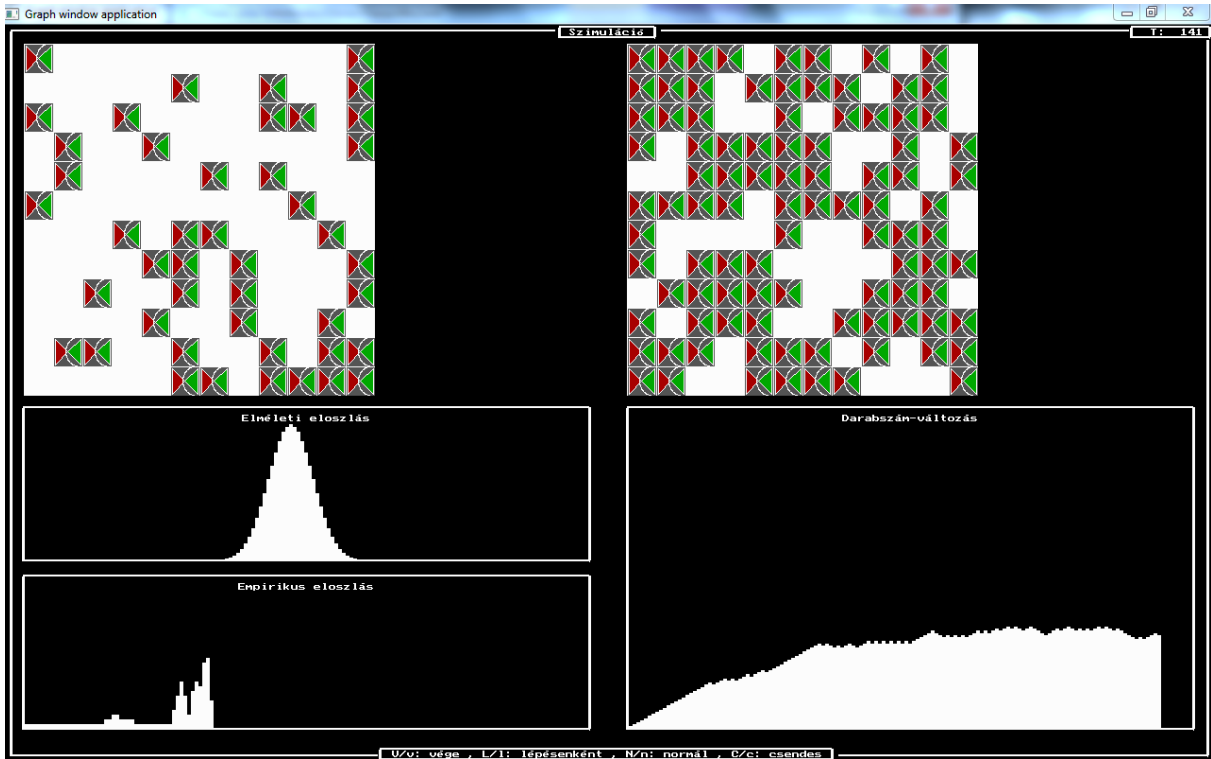
Ha N elemből áll az eloszlást reprezentáló sorozat (e_0, \dots, e_N) , akkor két ilyen távolsága legyen

$$\text{Távolság}(\text{emp}, \text{elv}) := \frac{1}{N} \sum_{k=0}^N (\text{emp}_k - \text{elv}_k)^2$$

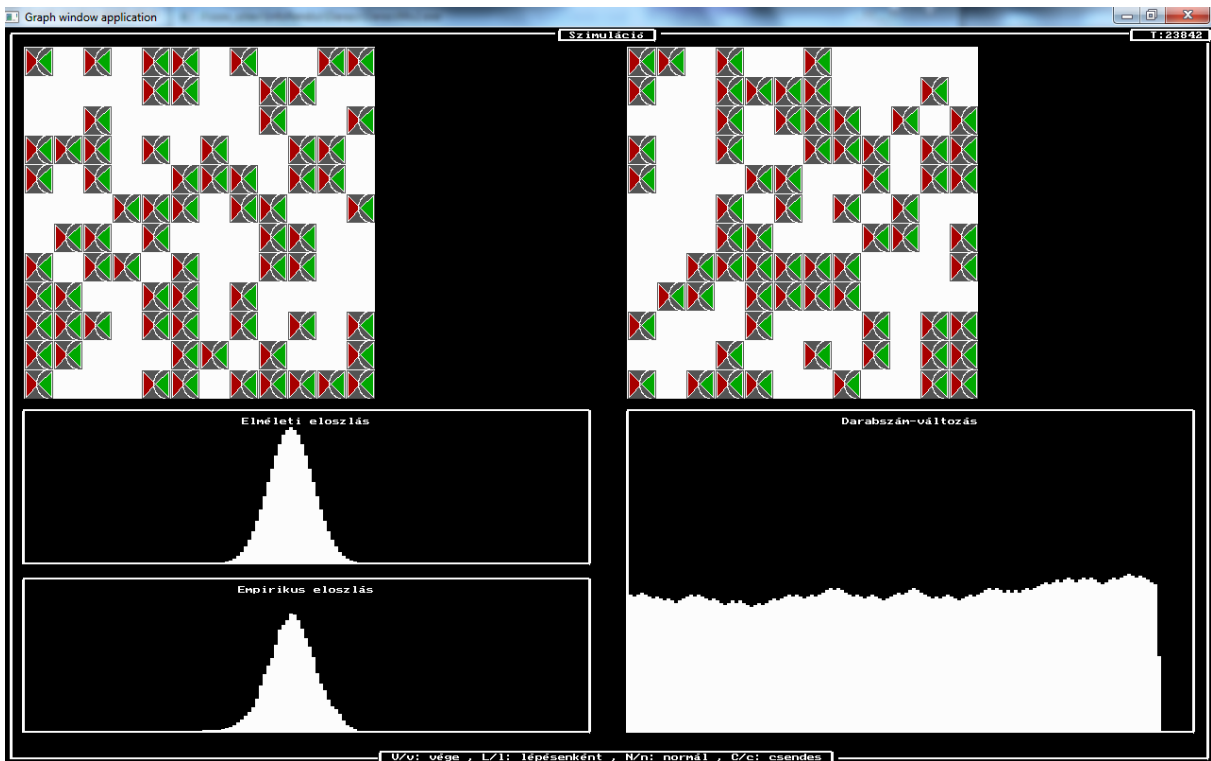
Hogy mire is gondolok? Álljon néhány jellemző kép az elérendő cél bemutatására!

² http://hu.wikipedia.org/wiki/Binomi%C3%A1lis_eloszl%C3%A1s

Darázs-modell

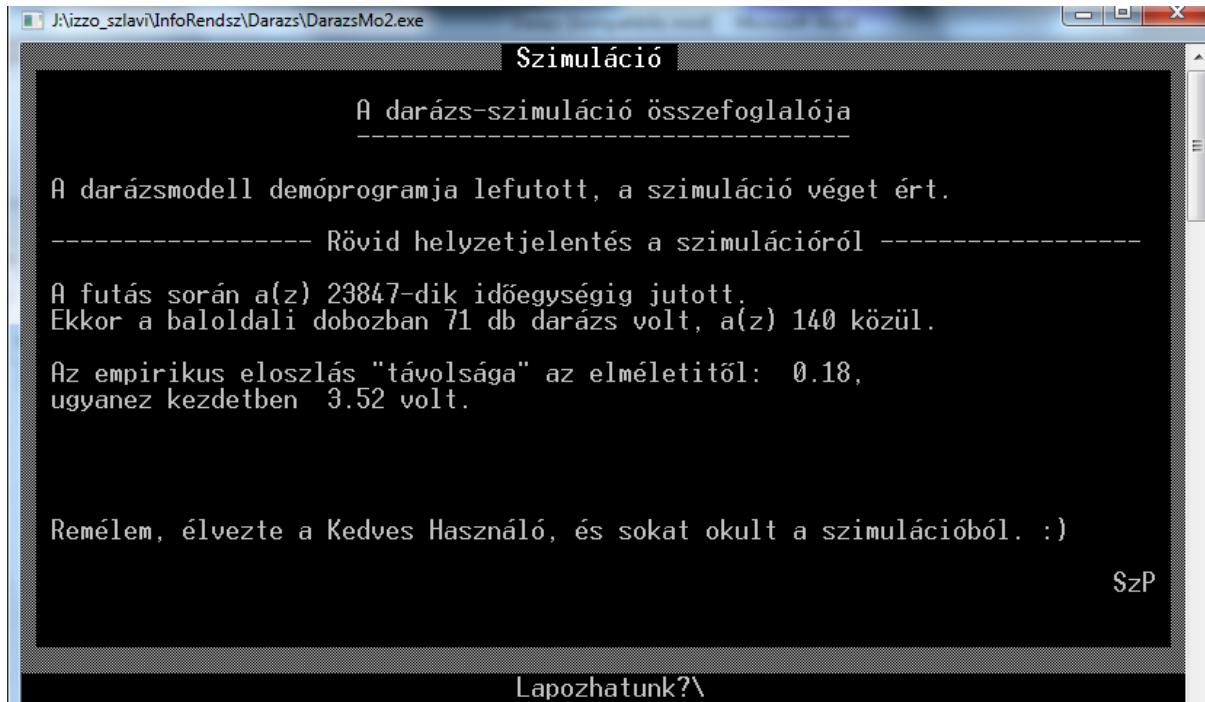


7. ábra. Egy futási kép a szimuláció elejéről (T=141)



8. ábra. Egy futási kép a futás legvégén (T=23842)

Darázs-modell



```
J:\izzo_szlavi\InfoRendsz\Darazs\DarazsMo2.exe
Szimuláció

A darázs-szimuláció összefoglalója
-----
A darázsmodell demóprogramja lefutott, a szimuláció véget ért.

----- Rövid helyzetjelentés a szimulációról -----

A futás során a(z) 23847-dik időegységig jutott.
Ekkor a baloldali dobozban 71 db darázs volt, a(z) 140 közül.

Az empirikus eloszlás "távolsága" az elméletitől: 0.18,
ugyanaz kezdetben 3.52 volt.

Remélem, élvezte a Kedves Használó, és sokat okult a szimulációból. :)

SzP

Lapozhatunk?\
```

9. ábra. Egy futási kép – Összefoglaló

5. A teljes anyag

Ez a kézirat, források, lefordított kódok, súgófájl... : [Darazs.zip](#).