

## 3D ábrázolás – PoVRay (8. rész)

A PoVRay által készített képek minősége arányos az elkészítésére (renderelésre) fordított idővel. Jobban mondva egy kép minőségét duplájára emelve a szükséges idő négyzetesen növekszik. Egy-egy kép esetén kivárható időket kapunk, azonban animációk esetén több száz képkocka elkészítése napokba is kerülhet. Mikor még tíz éve egy 486/66MHz gépen műveltem ezt a tudományt, egy 1024x768 felbontású kép másfél órán át készült.

### A felbontás változtatása

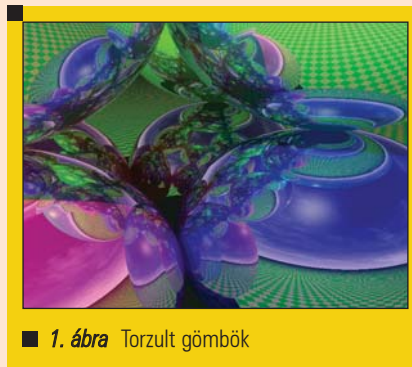
Egy kép minőségére leginkább a felbontása a jellemző, amelyet a pixelek száma határoz meg. A felbontással óvatosan kell eljárni, a vízszintes és a függőleges méret arányát a használt kamera határozza meg, ha eltérő az arány, akkor torzul a kép.

```
camera{
  up y
  right 1.0*x
  location <10,10,20>
  look_at <4,4,4>}
```

A kameránál a `right` kulcsszó adja meg az arány, amely szokványos képernyők esetén 1.333\*x, vagyis 4:3 az arány. Ha fotó formátumot szeretnénk, amely 3:2 arányú (például 15x10), akkor 1.5\*x kell a `right` kulcsszó után. A fenti példában a képformátum aránya 1:1, ha ezt 4:3 arányú képre készítjük el, akkor torzítva jelennek meg a gömbök (1. ábra), kissé laposabbak lesznek, hiszen csökken az ábrázolt magasságuk.

### Progresszív renderelés

Nagyobb projektek esetén célszerű progresszív renderelést választani, amely esetén a *PoVRay* először nagyobb négyzeteket összevonva jelenít meg, majd ezeket fokozatosan finomítja (2. ábra). Több perces elkészítési idő felett elnagyolva láthatjuk a végeredményt, s megszakíthatjuk a renderelést, ha nem megfelelő



■ 1. ábra Torzult gömbök



■ 2. ábra Progresszív renderelés

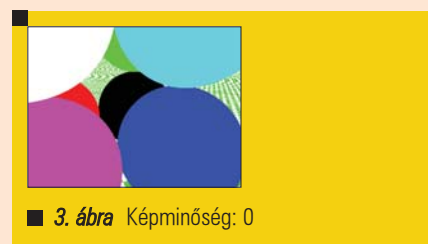
a tárgyak elrendezése vagy a kinézte. Általában a megfelelő kameraállás kikísérletezésénél jön jól. Az elkészült kép nem lesz progresszív (néhány képformátumnak van ilyen tulajdonsága). A progresszív módot az `SP`, mint a kezdő négyzet pixelmérete, illetve az `EP`, mint az utolsó progresszív négyzet pixelmérete határozza meg.

```
$ povray +w800 +H600 +P +EP2
↳ +SP32 last01.pov
```

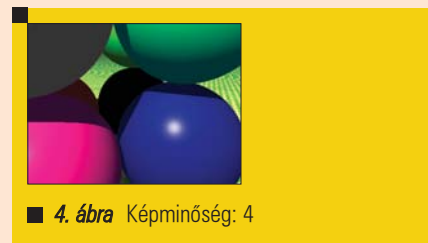
### A sugárkövetés minősége

Mivel a *PoVRay* sugárkövetéssel dolgozik, sok opciója van a minőség beállítására. Elsősorban a közvetlen képminőséget tudjuk befolyásolni, amelyet egy 0-9 közötti számmal kell megadnunk, ahol a 0 a legrosszabb képminőség (3-5. ábra):

```
$ povray +w800 +H600 +P +Q4
↳ last01.pov +o1ast04.png
```



■ 3. ábra Képminőség: 0



■ 4. ábra Képminőség: 4



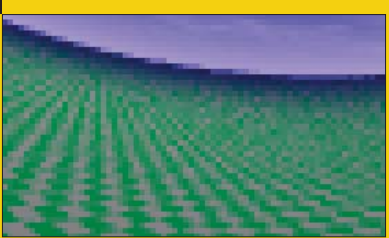
■ 5. ábra Képminőség: 9



6. ábra Élsimítás nélkül



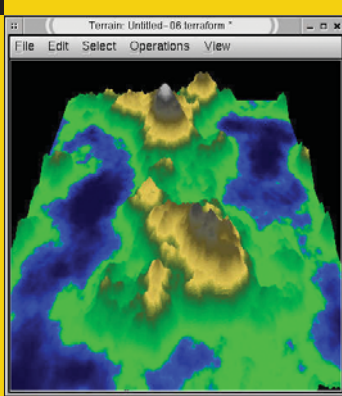
7. ábra Élsimítással



8. ábra Élsimítással



9. ábra Élsimítással és rekurzióval



10. ábra Terraform: hegyvidék

### Élsimítás

A raszteres – vagyis pixeleken alapuló – képábrázolás egyik legrosszabb tulajdonsága az egyenesek ábrázolása, ugyanis ezen esetben törések lesznek a pixelek határán. Az élsimítást egy 0 és 1 közötti számmal tudjuk beállítani, minél közelebb megyünk a nullához, annál jobb képet kapunk eredményül (s annál tovább tart a renderelés is).

```
$ povray +w800 +H600 +P +A0.05
↳ last01.pov +o1ast06.png
```

Ahogy a 6-7. ábrán is látszik, az élsimítás hatása, a 7. ábrán élsimítással ugyanakkora felbontáson a kép sokkal szebben néz ki, eltűntek a pixelek közötti durva színbeli ugrások. Ennél jobb minőséget érhetünk el, ha az élsimítást többszörös rekurzióval

kombináljuk, bár ezzel sokszorosára növekszik a kép elkészítésének ideje, de az eredmény kárpótol majd.

```
$ povray +w800 +H600 +P +A0.05
↳ +R20 last01.pov +o1ast09.png
```

Minél nagyobb számot használunk rekurzióval, annál jobb lesz a képünk minősége olyan részeken, ahol egy-két pixelnyi információt kell megjeleníteni: ilyen a kockás padló tükröképe. A két kép különbségéből látható, hogy a rekurzióval készített képen több részlet látszik, holott a felbontás azonos.

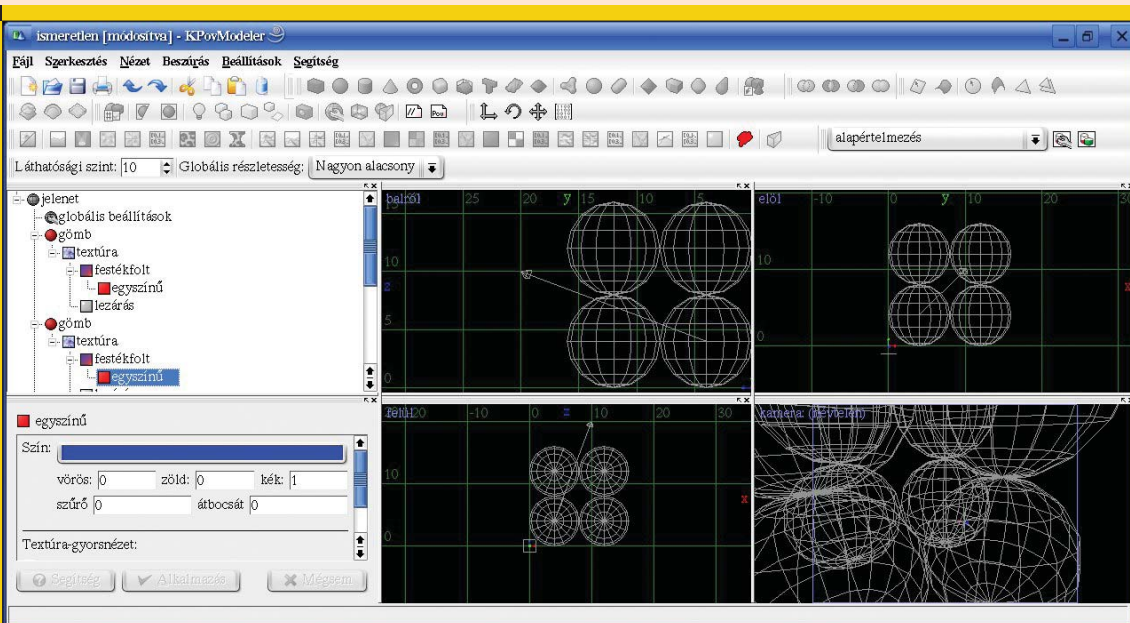
Egy komolyabb 3D világ renderelése – átlagos felbontással és erőteljes élsimítással akár egy-másfél óráig is eltarthat egy vadonatúj 3GHz-es gépen.

### Előtté programok

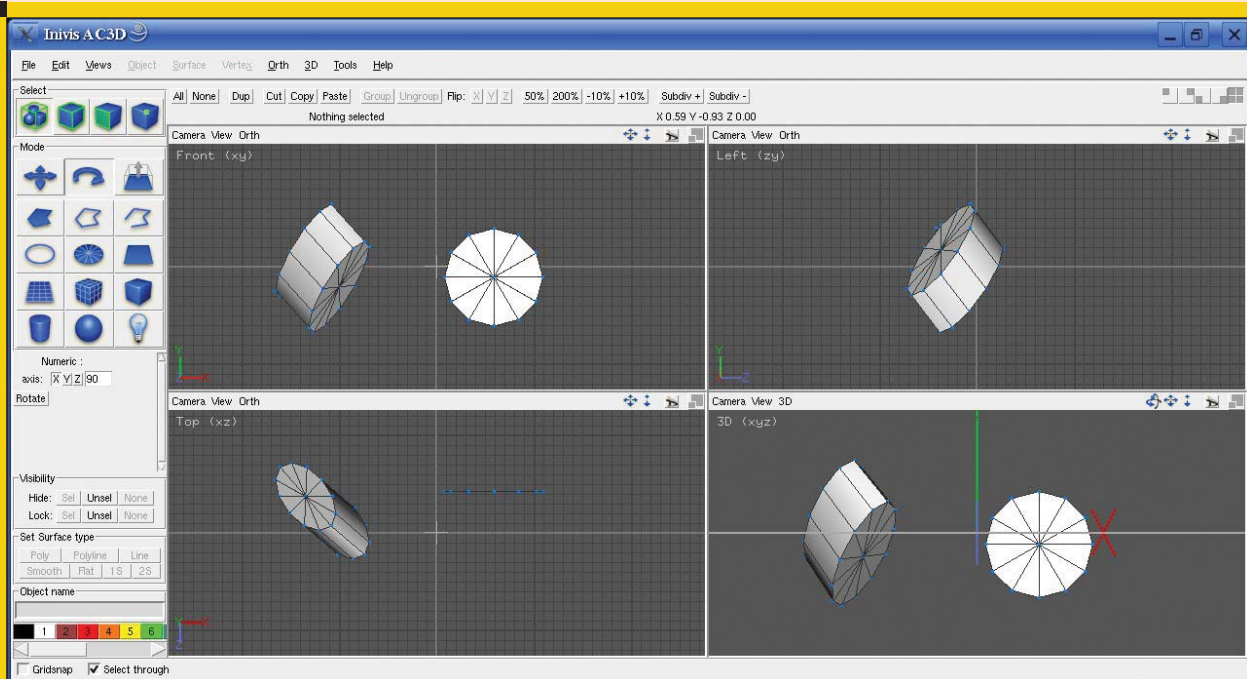
Többször is említettem, hogy a *PovRay* nem igazán alkalmas önmagában való használatra, hiszen a nyelvezete és a vizuális kontroll teljes hiánya nehézkessé teszi a 3D világ felépítését, az animáció pedig teljesen megöli a szárnyaló fantáziát. Programozói vénával nem rendelkező grafikus emberek pedig elborzadnak a gondolattól, hogy egy 3D ábrát szöveges felületen készítsenek el.

### Terraform

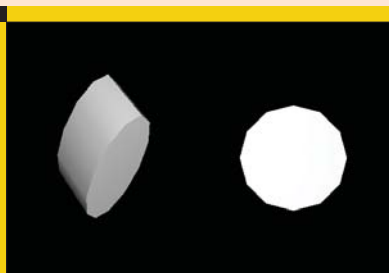
Az egyik kedvenc programom a *Terraform*, amellyel tájat lehetett készíteni, hegyeket, folyókat és tavakat,



11. ábra KPovModeler: a cikkben szereplő nyolc gömb



12. ábra AC3D: két poligon alapú tárgy



13. ábra AC3D: a PoVRay export renderelve

ezeket magasság szerint színezní, textúrázní. Sajnos az utolsó kiadása négy éves, bináris formában nem tudtam már feltelepítení, a lefordítást pedig pár órásh munká után szintén feladtam: a program egyszerűen olyan régi, hogy a mostani disztribúciókon már nem fut (az *OpenSuSE 10.2 Alpha1* pedig különösen friss). Remélem akad valaki, aki ezt a remek programot újracsatí.

### KPovModeler

A *Terraform* fejlesztésének befejezésével egy időben jelent meg a *KPovModeler*, amely egy átlagos 3D modelleket kezelni tudó program kinézetével bír. A fő munkaterület a 3D világ három nézetét (balról, előlről, felülről) és a kiválasztott kamera képét mutatja. A tárgyakat meg tudjuk fogni az egérrel, illetve egérműve-

letekkel átméretezní, áthúzní, elforgatní is képesek vagyunk. A tárgyaink egy fa struktúrában szerepelnek, amelyben ki tudjuk választani egyenként is a szükséges elemeket. A magyartíása kissé hiányos (illetve inkább érdekes :), ennek ellenére kis töprengéssel rá lehet jönní, hogy a „lezárás” szó alatt a *PoVRay* finish kulcsszóját értik. A program írni és olvasni tudja a POV állományokat, viszont eléggé szűk dialektust beszél: kizárólag a 3.5-ös *PoVRay* aktuális szintaxisát érti, s nem követi nyomon az include állományokat sem.

### AC3D

Az *AC3D* egy fizetős program, egy 14 napos próbaverziót tudunk megnézni belőle, tudása és használata kilóg a többi közül, elsősorban poligon alapú modellezésre készült, viszont képes exportálni *PoVRay* formátumba bármilyen elterjedt 3D CAD programból (*3D Studio*, *AutoCAD*, stb), illetve játékok állományiból (*Quake3*, stb).

### Zárszó

Őszintén remélem, hogy a *PoVRay* használatát bemutató kis sorozatom nem riasztott el senkit a mélyebb megismeréstől. A program jelentős megkötésekkel használható, főképp háttérként szolgál sok olyan progra-

mot segítve, amelyeknek írói nem akarnak bajlódni egy 3D modellező megírásával, viszont a *PoVRay* leíró nyelvét elsajátítva a programjuk képes lesz szemléletes ábrák készítésére és animálására (molekula modellezők, orvosi felhasználás, ipari berendezések animációi, stb).



**Auth Gábor**  
(auth.gabor@enaplo.hu)

Egy pécsi középiskolában informatikát és programozást oktat.

Tíz éve botlott először a UNIX rendszerekbe, 7 év Linux használat után kapta el a FreeBSD lázat, amiből máig nem tudott kigyógyulni.

### KAPCSOLÓDÓ CÍMEK

- A PovRay projekt honlapja  
➔ <http://www.povray.org>
- A cikkben említett fájlok  
➔ <http://user.enaplo.hu/~auth.gabor/pov/>
- Terraform  
➔ <http://terraform.sourceforge.net>
- KPovModeler  
➔ <http://www.kde.org>
- AC3D  
➔ <http://www.ac3d.org>