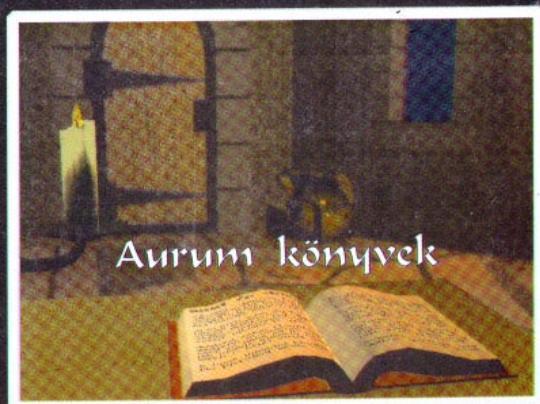


Arany Sándor

3D grafika és animáció  
Amigán és PC-n

# Imagine II.



Aurum könyvek

Arany Sándor

3D grafika és animáció  
Amigán és PC-n

# Imagine II. kötet

AURUM  
Könyvek #5

**Aurum Könyvek sorozat #5**  
Sorozatszerkesztő:  
Arany Sándor

**3D grafika és animáció  
Amigán és PC-n  
Imagine II. kötet**

© Arany Sándor 1994

**ISSN 1218-2761  
ISBN 963 04 4341 4 Ö  
ISBN 963 04 5517 X II.K.**

Kiadja az AURUM DTP Stúdió  
Tiszaföldvár, Ószőlő Fő út 64.  
Felelős kiadó: Arany Sándor  
A szerkesztés Amiga 4000 DTP rendszeren,  
PageStream tördelőprogrammal készült

Borítóterv: Aurum

Nyomdai munkálatok:  
Kollányi Nyomda Tiszaföldvár

## Bevezető

Régi adósságom rovom le ezzel a könyvvel, közel egy éve, az első kötet megjelenésekor már ígértem a folytatást. A kötet teljesen az Imagine 3.2-es verzió újdonságaival foglalkozik, amit már a 2.0-s változat is tudott, arról itt nem esik külön szó, hiszen ezen funkciók leírása fellelhető az előző kötetben.

A 3.2-es változat igen sok lényeges újítást hozott elődjéhez képest, ezek nagy része az animációk készítésével kapcsolatos. A tárgyakat csontvázak segítségével is mozgathatjuk. A csontváz deformálása automatikusan átkerül a tárgyra. *(A csontváz lehet Skeleton, Skeleton Giant, Skeleton Warrior, de itt-ott alkalmazhatunk Zombikat is, ha elég régiek és kilátszik a csontvázuk.)* (Ne hallgass erre az esze-hagyottra kedves olvasó, Ő mindig beleszól mindenbe, pedig semmi köze a dolgokhoz. Sajnos azonban el kell tűrnöm, mer Ő is én vagyok. Orvos mondta nem annyira veszélyes, ha nem ellenkeznek vele. Azért majd titokban zárójelezem, hátha azt nem veszi észre.) Ezzel a funkcióval könnyedén készíthetünk, olyan, eddig nehézkesen létrehozható mozgásokat, mint egy ember járása, vagy a madarak röpte.

A tárgyak alak és tulajdonságváltozásait immár magában a tárgyban tárolhatjuk, hasonlóan a cycle objectekhez, azonban ezek a változások nem kell, hogy ciklikusak legyenek, az egyes mozgásfázisok felhasználásáról az animáció beállításakor az Action editorban dönthetünk. Két fázis között az átmenetet automatikusan kiszámolja a program.

Egy új szerkesztővel is bővült a program, a Spline editorral. Ebben görbe vonalak által határolt tárgyakat szerkeszthetünk. Itt a tárgyakat nem a szokásos egyenes vonalak határolják, hanem kontrol pontokkal irányított görbék, így elkerülhető a tárgyak szögletesége. Munkánk során felhasználhatunk Adobe Type 1

## Bevezető

---

formátumú fontokat is. A spline tárgyakat a renderelés előtt azonban át kell alakítani poligonális formába, de még így is jobb eredményt érhetünk el, mintha eleve vonalakból állítottuk volna elő azokat.

Az Action editor, mint már említettem sok animáció készítést megkönnyítő funkcióval bővült, a morfózisok készítése, a szerkezeti mozgások beállítása sokkal egyszerűbbé vált. Az animációs effektusok száma jelentősen bővült, sőt megjelent egy új típus, a Global F/X, amelybe globális hatású effektek tartoznak. Ezek hatásukat nem egy tárgyra, hanem a Globals-ra fejtik ki, a kép egészét módosítják. Ilyen effektus például a Toon, amelynek hatására a renderelt kép úgy néz ki, mintha rajzolva volna.

Új hatás a programban a mélységélesség generálása. A beállított fókusz távolság előtt és mögött lévő tárgyak a távolság függvényében életlenebbül látszódnak, mint ahogy a fényképeken.

A tárgyakon használható textúrák száma öröndetesen megszaporodott, száz fölé emelkedett. Közöttük vannak olyanok, amelyek a kialakított mintázatot animálják. Ilyen például a tüzet utánozó Fire, vagy az esőáztatta vízfelületet modellező Rain.

Van egy különleges textúra csoport, a Light textúrák csoportja, amelyeket fényforrásokra alkalmazva érdekes fényjátékokat érhetünk el.

Az új textúrák és effektek részletes leírása mellett a könyvben található egy színes melléklet, amely egy-egy kis képpel bemutatja az elérhető mintázatokat, megkönnyítve ezzel a megfelelő textúra kiválasztását, hiszen ennyit fejben tartani szinte lehetetlen.

A Cycle és a Forms editor semmit nem változott az előző könyv megjelenése óta, ezért most nem is esik róluk több szó.

Nos, ennyi könnyed áttekintés után vágjunk is bele a sűrűjébe. Kezdődjön a Detail editor újdonságainak részletezése.

## Detail editor

A Detail editor kinézetre semmit nem változott, de jó néhány hasznos funkcióval bővült. Lényeges újítás a State animáció támogatása, amikor is az egyes mozgásfázisokat és tulajdonságváltozásokat magában a tárgyban tároljuk el, ezzel a morfózikus animációk készítését jelentősen megkönnyíthetjük, mert nem kell több tárgyat létrehozni az egyes részállapotokhoz. Másik fontos újítás a csontvázkontrol, vagyis amikor a tárgy alakját csontokkal, vezérvonalakkal módosítjuk.

A csontvázkontrolhoz hasonló dolog az inverz kinematika, amivel úgy manipulálhatjuk a csoport elemeit, mintha a csoportbakötést jelképező vonal fizikailag is összekapcsolná az elemeket. A csoport egyes tagjait mozgatva ezeken keresztül átadódik a mozgás a többi elemre. Ezt kihasználva könnyedén készíthetünk olyan animációkat, mint egy főtengely forgása, vonatszerelvény, stb.

A tervezést jelentősen könnyíti, hogy a perspektíva ablakot függetlenítették a három másik szerkesztő ablaktól, így bármilyen irányból és távolságból szemlélhetjük a tárgyakat. A másik új dolog a perspektíva ablakban, hogy akár ott is manipulálhatjuk a tárgyakat. Még egy újítás tartozik a perspektivikus nézethez, kijelölhetünk rajta egy zónát a Quickrender számára. Ezután csak ennek a zónának a tartalmát számítja ki a QR, ami jelentős gyorsulást jelent kis részletek ellenőrzésénél.

Mind a négy szerkesztő ablak háttérébe tetszőleges képet tölthetünk be (természetesen az Imagine képernyőjének megfelelő felbontásban), amellyel a tárgy alakjának meghatározása válik könnyebbé.

## Detail editor

---

Eddig, főleg PC-n, ahol hiányzik a multitaszk, nehézkes volt a megfelelő brush kiválasztása, most a programba integrálva találunk egy kép és egy animációnéző programot.

Az interaktív műveletek száma megnövekedett, a mozgatás, forgatás és méretezés mellett különböző tárgytorzításokat is végezhetünk valós időben, interaktív kontrol mellett.

A munka kényelmesebbé tételére eddig is használhattunk User Gadgeteket, erre most két sornyi lehetőség van.

Egy érdekes új funkcióval a tárgyakat részecskékre szedhetjük, például piciny golyókból épülhet fel az object. A részecskék alakja, mérete szabadon definiálható.

Különösen az **AutoCAD** és az azzal kompatibilis programok használó számára hasznos, hogy az **Object/Load DXF** és **Save DXF** menüpontokkal közvetlenül tudják a **DXF** formátumú fájlokat felhasználni.

A 25 leggyakrabban használt attribútumot külön csokorba gyűjthetjük és ezek bármelyikét könnyedén, az attributes kérdezőbe való belépés nélkül, hozzárendelhetjük tárgyainkhoz.

Egy új funkció segítségével csatlakozó tárgyak közé könnyedén tudunk felületeket képezni. Hasonló ez a funkció a régebbiről ismert Skin-hez, azonban sokkal többre képes, mert az összekapcsolandó elemeknek nem kell azonos számú és sorrendű pontokból állnia.

A rejtett, takarásban lévő pontok elérése és szerkesztése egyszerűsödött, mivel a **Hide** mellett lehetőség van **UnHide**-re, ráadásul a kitakarandó, vagy újból megjelenítendő pontok megadása egyszerűbbé vált, azonosításukra használhatjuk a subgroupokat.

Miután nagy vonalakban áttekintettük az újdonságokat, ideje hogy részletesen is megismerkedjünk használatukkal.

### Új megjelenítési lehetőségek

Az itt leírandó funkciók és lehetőségek azonosak a program többi tárgyszerkesztőjében is, ezért leírásuk egy helyen, itt található.

A tárgyak attribútumainak beállítása során jól jöhet, ha a programból való kilépés nélkül is megnézhetünk egy-egy brushnak szánt képet. Erre szolgál a **Project** menüből a **Show Pic** funkció. Segítségével egy fájl szelektoron keresztül választhatjuk ki a képet. A formátum Amigán **IFF-ILBM**, **PC-n TIFF**, **Targa**, **PCX**, vagy **GIF** lehet, összhangban a használható brush formátumokkal.

Hasonló célt szolgál a **Show ANIM** menüpont, amely animációk megtekintését segíti. Az animációk formátuma **IFF-Anim5**, **FLC**, vagy **FLI** lehet. Ez a funkció is igen hasznos, hiszen ebben a programváltozatban az animált brushoknak nem csak képsorozatokot, hanem akár összefűzött animációkat is használhatunk.

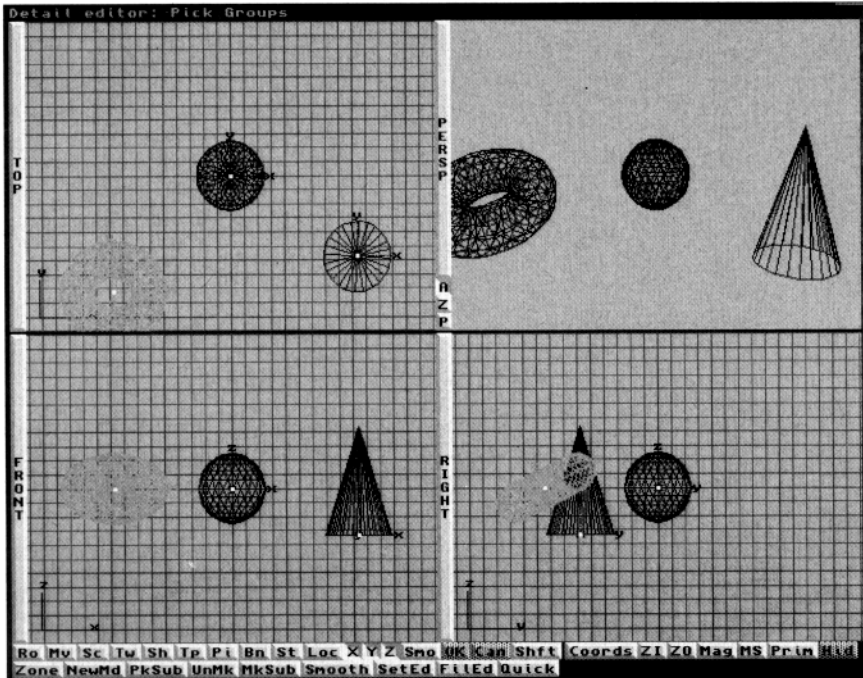
A project menü harmadik új pontja a **Memory**, amellyel a rendelkezésre álló memória méretéről szerezhetünk információt.

A tárgyak perspektivikus megjelenítése sok újdonságot hozott. Egyike ezeknek, hogy a perspektíva nézet függetlenül lett a síknézeti ablakoktól, bármilyen irányból és távolságból tekinthetünk immár a szerkesztett tárgyakra. Ennek megvalósítására a régi két forgató gomb eltűnt a perspektíva ablakból, helyette megjelent három kapcsoló, melyekkel kiválaszthatjuk milyen műveletet akarunk végrehajtani a nézeten.



## Detail editor

A három kapcsoló közül az első, amelynek "A" betű a jele, a szerkesztő tér forgatására szolgál. Az egér gombját a perspektíva ablakban lenyomva, majd az egeret elmozgatva, tetszés szerinti nézőirányt állíthatunk be.



A "Z" kapcsolót benyomva az előzővel azonos módon a megfigyelési távolságot szabályozhatjuk.

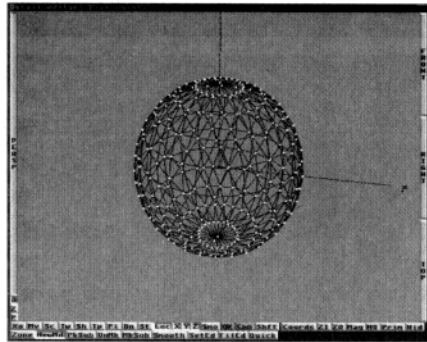
A harmadik, a "P" gomb benyomása után a látószöget, vagyis a képzeletbeli kamera gyújtótávolságát lehet beállítani. A gyújtótávolság szoros összefüggésben van a perspektivikus hatással. Ha a látószög kicsi (teleobjektív), a térhatás kevésbé alakul ki. Nagyon széles látószögű optikát használva torz perspektívát kaphatunk, amely bizonyos

esetekben előnyös is lehet. Egészen széles látószög esetén jön létre a "halszemhatás".

Utóbbi két paramétert numerikusan is megadhatjuk a **Display/Perspective** alpontjai segítségével. A **Reset** alaphelyzetbe állítja a perspektíva nézetet. A **FOV angle** a kamera nyílásszögét jelenti. Ennek értéke 1-150° között változhat, ha az egerrel a perspektíva ablakban állítjuk be, de numerikusan max 179°-s látószöget adhatunk meg. A **Zoom ratio** a nézet zoomolásának aránya a síknézethez képest. Ha a síknézetet közelebb hozzuk, a perspektíva nézet is ilyen arányban kerül közelebb.

A perspektíva ablakhoz kapcsolódó másik új lehetőség, hogy szerkeszthetjük benne a tárgyakat, ugyan úgy, mint a síknézeti ablakokban. Ehhez a **Display/NewMode** pontot kell kipipálni. Hogy a tárgyakhoz hozzáférjünk, az előbb említett három kapcsoló egyikének sem lehet bekapcsolva lenni, mert akkor a neki megfelelő művelet hajtódna végre. A **NewMode** lehetőséget ad a tárgyak térhatásának pontosabb kialakításához, hiszen a szerkesztés során rögtön látjuk is ennek a változását.

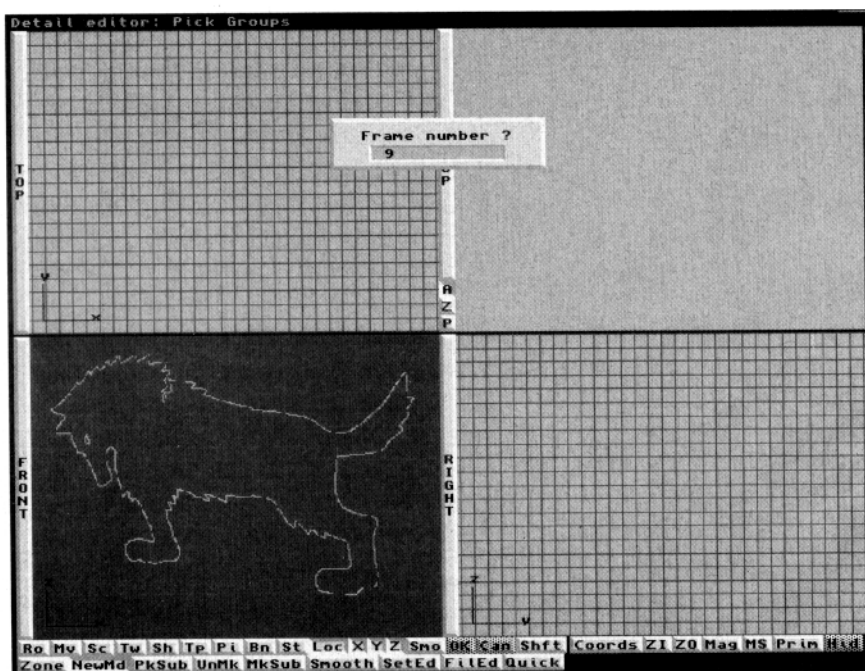
Ha egy tárgynak csak egy kis részletét szeretnénk megvizsgálni, hogyan fog kinézni a kész képen, nem kell az egészet lerenderelni, kijelölhetünk a perspektíva ablakban egy tetszőleges téglalap alakú területet és csak ennek a tartalmát számolja ki a program, amely természetesen jelentős időnyereséggel jár.



## Detail editor

A zónakijelölés felettébb egyszerű művelet, a **Display/Set Zone** pontjának kiválasztása után a perspektíva ablakban az egér gombjának nyomva tartása mellett meg kell húzni a zóna átlóját. A zónát bekerítő gumiszalagot addig látjuk, míg a gombot el nem engedjük. A **Set Zone** ismételt használatával bármikor új zónát jelölhetünk ki, a **Clear Zone** menüponttal pedig törölhetjük azt. Míg az utóbbi műveletet nem alkalmazzuk, a zóna él, sőt, még a project editorba átlépve is megmarad, ahol a rendering előtt eldönthetjük, használjuk-e.

A megjelenítési lehetőségek utolsó újítása, hogy a szerkesztőablakok háttérébe tetszés szerinti képet tölthetünk. Ez a kép természetesen csak a képernyő szinkészletével jelenik meg, az ablakhoz igazított méretben.



Mind a négy ablaknak saját háttere lehet, az új hátér mindig az aktuális ablakba kerül. Hátteret betölteni a **Display/Load Backdrop** pontjával lehet. A **Clear Backdrop** törli az összes hátteret. Nem csak önálló képet tölthetünk háttérbe, hanem animációt is, illetve egyszerre csak egy kockáját. Az animáció Amigán **IFF-Animbrush**, PC-n **FLI** lehet. A fájl kiválasztása után ekkor azt is meg kell adni, hogy az anim hányadik képkockája töltődjön be.

A hátterek használata megkönnyíti a tárgyak szerkesztését, mert így pl. a tetszőleges alakú tárgyak körszerkeszthetők.

A Display menüben található még egy menüpont, amelyről nem esett szó, ez a **Reset Views**. Hatására mind a négy szerkesztőablak alapállapotba kerül.

### Régi funkciók új helyen

Említettem, hogy a program régi változatában már meglévő funkciók leírásával most nem foglalkozom, mégis ejtsünk néhány szót azokról. Némely menüpont a jobb szervezés érdekeben "menüt váltott", azaz máshová került. Szerencsére a billentyűzet rövidítések változatlanok maradtak, így akik ezeket használták, semmit sem vesznek észre a cseréből. Azoknak, akik a funkciókat a menükből hívják, elejében zavaró lehet a csere, de rövidesen megszokják és belátják, jobb ez így. Miket érint a változás? Néhány, a tárgykezeléshez tartozó funkció került át a **Functions** menüből az **Objects**-be, például a **Fracture**, valamint a **Mold** kérdezőjéből a nem tárgykinyomás témakörébe tartozó **Conform to ...** funkciók új menübe, a **Functions/Conformations**-ba kerültek át. Az **Attributes** az **Objects** menüből a **Functions**-ba lett helyezve.

### State-ek

Mi az a State? Ez a hármas Imagine legnagyobb és leghasznosabb újítása, segítségével egy kényelmes animációs lehetőséghez jut a felhasználó. A State állapotot jelent és a programban is ilyen értelemben van használva. Lényege, hogy a tárgyról több kulcsállapotot tárolunk el, majd az animáció során az egyes kulcsállapotok közötti részeket a program számítja ki. Hasonlatos a dolog a cycle objectekhez, ebben az esetben is maga a tárgy tárolja a kulcsállapotokat. Van azonban két lényeges különbség:

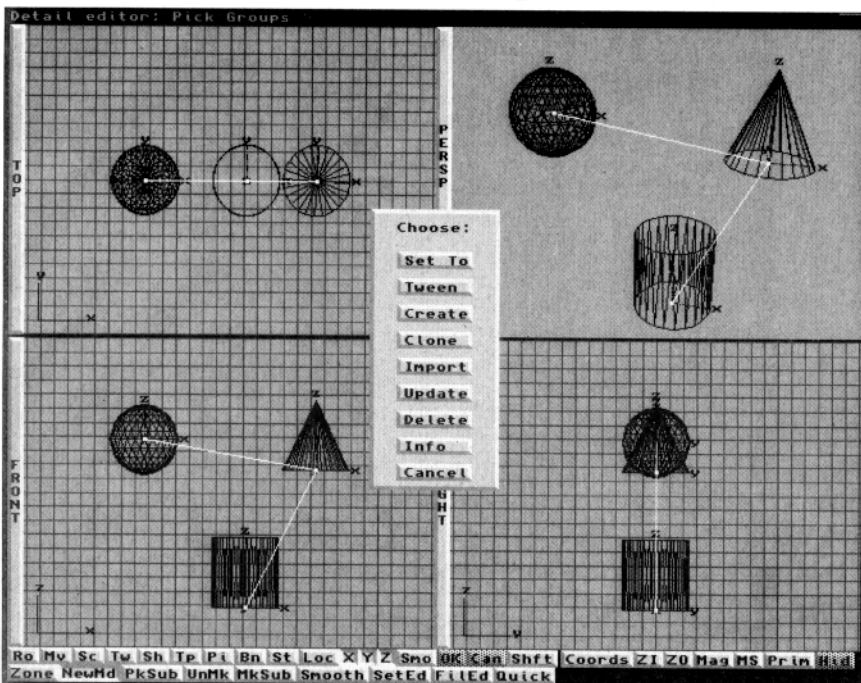
- a., Nem csak az egyes csoportelemek egymáshoz viszonyított helyzete tárolható el, hanem az összetevő tárgyak alakja, színe és egyéb attribútumai, vagy a rájuk alkalmazott textúrák és brushok adatai is.
- b., Az egyes állapotok nem csak a készítésük sorrendjében, hanem tetszőleges sorrendben is felhasználhatók, az állapotok többször sorra kerülhetnek a ciklus során, azonosításukra 17 karakteres név szolgál.

Látható tehát, hogy a State animáció jóval szabadabb lehetőségeket nyújt, mint a Cycle editor, gyakorlatilag feleslegessé is teszi azt.

A legjobb lesz, ha további elméleti fejtegetések helyett kipróbáljuk ezt a módszert. Hozz létre három egyszerű tárgyat, mondjuk egy gömböt, kúpot és egy hengert. Kapcsold össze azokat egy csoportba, ahogy az a következő képen látszik.

Erről jut eszembe, a **Group** és **Ungroup** funkciók átkerültek a **States** menübe, itt keresd.

Ez lesz a kiindulási állapot, készíteni kell belőle egy state-et. Válaszd ki az egész csoportot, majd aktiváld a **States/States** menüpontot.



Megjelenik egy kapcsolótábla kilenc bigyulával, ezek közül a Cancel gondolom nem szorul magyarázatra.

Bökj a **Create** feliratúra, megjelenik egy input mező, ahol a State nevét lehet megadni. Ha még nincs state definiálva, a program felajánlja a "**Default**" nevet. Érdeemes ezt elfogadni, később látni fogjuk miért. A további state nevek tetszőlegesek lehetnek.

Ne örülj, nincs még vége, ismét egy kapcsolótábla következik. Ebben olvashatod az imént adott nevet (meg is változtathatod). Alatta figyel öt kapcsoló,

## Detail editor

---

ezekkel lehet meghatározni, hogy az adott state a tárgy milyen típusú állapotait tárolja el. Ezek közül egyszerre több is aktiválható, sőt, akár mind is. Vigyázni kell azonban, mivel az adatok eltárolása helyet igényel, a fájl méretére könnyen alkalmazódik a rétestészta effektus!

A következő adattípusok menthetők el:

**Shape** - A tárgy alakja. A hagyományos értelemben vett morfózis alapja.

**Face Colors** - A tárgy egyes felületelemeinek **Color**, **Reflect** és **Filter** értéke kerül eltárolásra. Ez csak az egyedi attribútumokat jelenti, amelyeket Pick Faces módban adtunk a kijelölt felületeknek. Nem érinti a globális, egész tárgyra adott attribútumokat.

**Grouping** - A csoport egyes tagjainak helyzete, mérete, orientációja.

**Object Props** - A tárgy attribútumai. Ebbe csak a tárgy egészére adott tulajdonságok tartoznak, a Pick Faces módban módosítottak nem. A tárolt attribútumok: **Color**, **Reflect**, **Filter**, **Specular**, **Dithering**, **Hardness**, **Roughness**, **Shininess**, **IofR**, **Fog Lenght**, **Phong**, **Bright**, **Fastdraw**, **QuickEdges**, **Light**. Nem tárolódnak a textúrák és brushok adatai.

**Textures/Brushes** - A tárgyra felvitt textúrák és brushok paraméterei. Az animált tetúrák készítéséhez nélkülözhetetlen.

Kapcsold most be a **Group** kapcsolót. Klicnk on to OK.

Következő lépésben a csoport utolsó elemét mozgasd el eredeti helyéről. Ha ez megvan, ismét válaszd ki a teljes csoportot, majd készíts egy új statet, mondjuk "kileng" néven. Most is a Group kapcsolót kell aktiválni.

A fenti műveletsorral elkészítettünk egy egyszerű state animáció alapjait. Az animációnak két kulcsállapota van. Hogyan lesz ebből animáció? Egyelőre csak a Detail editorban készíthető preview elkészítését részletezem, a végleges animáció beállításának mikéntjére az **Action editornál** derül fény. Állítsd úgy be a perspektíva nézetet, hogy a tárgyak mozgása majd jól látszódjon.

Tehát. Jelöld ki a csoportot, majd a **States** menüből aktiváld a **State Anim/Make** pontot. Eredmény egy figyelmeztetés, ami ebben az esetben nem hibára hívja fel a figyelmet, hanem az animáció beállításának mikéntjét meséli el. Ne foglalkozz vele, elmondom én magyarul, részletesebben. Szóval klikk on to OK.

Hopp, itt egy lista, benne a statek nevei. Válaszd ki azt, amelyiket kiindulási állapotnak szánod, mondjuk legyen ez most a Default (ez nem kötelező, mint kicsit korábban írtam, a felhasználás sorrendje tetszőleges és ez vonatkozik a kiindulási state-re is). A megjelenő input mezőben add meg, hogy az előbb választott állapotból a program hány képkocka alatt változtassa át a tárgyat a következő állapotba. Az enter után ismét az előbbi lista jelenik meg, válaszd most ki a másik állapotot. Ennek is meg kell adni az átalakulás hosszát. Ismét lista, klikkelj megint a Default state-re, de az átalakulásnak adj nullát. Ezzel jelenti, hogy befejeztük az előzetes anim beállítását, már csak azt kell eldönteni, hogy a tárgyakat milyen részletességgel vázolja fel a program. A **Full Objects Only** teljes drótvázat jelent, az **Use Objects Settings** a tárgyakra esetlegesen beállított QuickDraw szerinti ábrázolást, míg a **Bounding Boxes Only** befoglaló dobozos megjelenítést. Ha ezt is megadtad, elkezdődik az animáció kiszámítása.



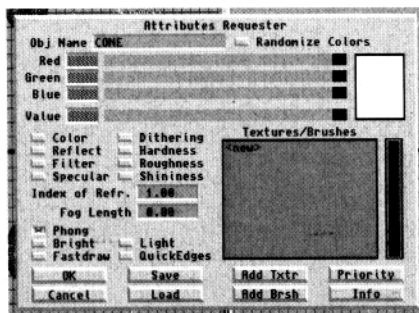
## Detail editor

Miután ez is megtörtént, a **States/State Anim/Play** segítségével meg is lehet tekinteni a művet. Az előzetes elkészítésére és megtekintésére egyéb iránt ugyan azok érvényesek, mint amit a Stage , vagy a Cycle editornál tapasztalhattunk.

A tárgy alakjával, vagy attribútumaival végzett manipulációk ugyan így tárolhatók el, és használhatók fel az animációk során.

Az elv az előzőekkel azonos, mégis érdemes néhány szót külön is szólni a textúra és brush paraméterek változtatásával készített animációkról. Ezek hatását nem tudjuk megtekinteni a Detail editorban készített previewen, ezért aki az elv mellett most azonnal kíváncsi a végeredményre is, lapozzon előre az Action editort leíró részhez, ahol a state object felhasználásának módja is olvasható.

A kísérletünkhöz elegendő egyetlen tárgy. Jelöld ki, majd lépj be az **Attributes** kérdezőbe (Új helyen, a Functionsban található!). A megjelenő kérdező kissé más, mint régen, de gondolom megbirkózol vele. Ha mégsem, lapozz kicsit előre, oda, ahol ennek a leírása található. Bökj az **Add Txtr**



kapcsolóra és válassz ki egy textúrát, mondjuk a Wood-ot. A paramétereit hagyd alapon, klikkelj tehát az OK-ra. OK-zd el az Attributes-t, majd készíts egy statet, amelyben a **Textures/Brushes** adattípust tárolod el. Vissza az Attributes-be. A kérdező textúra listájában válaszd ki az előbb létrehozott Wood-ot, majd bökj az **Info** gombra. Újból megkapod a Wood textúra paraméter kérdezőjét. Változtasd meg valamelyik értéket,

mondjuk a mintázat színét. Újabb state, ebben szintén a **Textures/Brushes** adatokat kell tárolni. Ha most készítenénk egy animacit az előbb létrehozott tárggyal, a faerezet színe folyamatosan változna (megfelelő beállítás esetén). Ugyan így járhatunk el, ha brusht feszítünk a tárgyra, annak méretét, helyzetét, orientációját, stb menthetjük le egy-egy statebe.

Fontos azonban tudni, hogy használt textúrák és brushok minden stateben azonosak, pl. nem lehet olyant csinálni, hogy az egyik stateben csak Wood textúrát használunk, a másokban csak Angular-t. Amikor egy meglévő textúrát/brusht törölünk, vagy hozzáadunk a tárgyhoz, az minden stateben törlődik, vagy létrejön. Új textúra/brush hozzáadásakor az éppen beállított értékekkel kerül be a többi state-be is (már ha egyáltalán van több state).

Másik hasonló korlátozás a brush kapcsolóinak eltárolására vonatkozik. Mivel ezek logikai igen-nem értékek, nincs köztes állapotuk, a morfózis során nem fokozatosan, hanem egyik kockáról a másikra változtatják értéküket. Például az egyik state-ben Flat-Flat feszítést definiálunk, a másokban Wrap-Wrap-ot. Nincs azonban olyan felfeszítési mód, amely Flat ugyan, de azért már egy kicsit Wrap, ezért az animáció során hirtelen fog változni a felfeszítés módja, akkor, amikor az új módot leíró state-t alkalmazzuk.

Statet létrehozni immár kiválóan tudunk, de mit kezdjünk a feleslegessé váltakkal? Töröljük azokat, ezáltal csökken a fájl méret. Statet törölni a **States/States** kérdező **Delete** kapcsolójával lehet. Megjelenik egy olyan lista, mint amivel a state anim preview készítésénél találkoztunk, ebből kell a halálraítélt nevét kiválasztani. Ezzel még nincs vége, a Delete nem csak úgy Delete, hogy puff, oszt vége!

## Detail editor

---

A program körültekintően megkérdezi, hogy melyik adattípust kívánod törölni, vagy netán az egész state-t. Előfordulhat ugyan is, hogy a Shape adattípusra már nincs szükség, de a Groupingra még igen. Ebben az esetben magát a state-t meghagyjuk, csak a feleslegessé vált adatokat dobjuk el. Olyan is lehetséges, hogy a state nem tartalmaz adatokat, csak van. Ez azt jelenti, hogy a state létezik, de egyetlen kapcsolója sem aktív. Ha egyáltalán nincs szükség a state-re, az **Entire State**-t kell kiikszelni.

Ez már igen, tudunk teremteni és pusztítani, kell ennél több? Igen, van még ezen kívül még néhány hasznos dolog. Csak úgy informatíve megkérdezem, mi van akkor, ha egy meglévő state adatait módosítani kell, mert mondjuk időközben rájöttünk, hogy nem is piros színbe kell változnia a bigyónak, hanem zöldbe? Készíthetünk új state-t a megváltozott adatokkal, de ez meglehetősen lamátor megoldás, kell hogy legyen valami jobb is. Például megváltoztatjuk a megfelelő adatokat, majd tudatjuk a programmal, hogy ezentúl a pirosbigyó nevű state ezeket tartalmazza. Igen ám, de itt jön egy új probléma, ha van már cirka öt tucat state, ki emlékszik arra, hogy a pirosbigyónak mik voltak a paraméterei, amelyeket meg kell változtatni? Nem ki, hanem mi, maga a tárgy. Ha szépen megkérjük, újfent felveszi a pirosbigyó nevű state-ben tárolt paramétereit. Azok az értékek, amelyekről az adott state nem rendelkezik, nem változnak meg. Például, ha az említett state csak az attribútumokat tárolja, az alak változatlan marad erre a state-re állva. Nosza States/States, kapcsolók, ahogy szokott, változatosság kedvéért most **Set To**. Ismét a jó öreg state lista lép színre, most azonban azt választhatjuk ki belőle, hogy mely state adatait vegye fel a

tárgy. Na persze ezt sem kell elsietni, csakúgy mint a kivégzést, hisz nem biztos, hogy mindent meg akarunk változtatni. Az adattípus kérdezőben megválaszthatjuk, hogy mely adattípusok state-ben tárolt értéke kerüljön a tárgyra. Ennek akkor van jelentősége, ha például a pirosbigyó state a szín mellett az alakot is tárolta, de utóbbit megváltoztattuk és szeretnénk, ha így is maradna, viszont az eredeti színre szükségünk van. Ekkor a Shape kapcsolót kikapcsolva, csak az Object Props adatai alkalmazódnak. A program nem enged olyan adattípust bekapcsolni, amely nincs tárolva az adott state-ben, mivel ennek nincs is értelme.

Ha beállítottunk mindent, jöhet a következő lépés, az updateolás. Erre szolgál a States kérdezőjében az **Update** feliratú gadget. A követendő művelet sor a szokásos, először kiválasztjuk, hogy melyik state-t szeretnénk updateolni, majd megadjuk, hogy mely adattípusok vegyék fel a tárgy jelenlegi nekik megfelelő paramétereit. Ezzel csak a már meglévő adattípusok értékeit lehet frissíteni, új csoportot nem lehet létrehozni.

Erre is van azonban lehetőség, csak ekkor nem az Update-t, hanem az **Info**-t kell kiválasztani. Lista, state név, és ismét itt a kis kérdező, az adattípusok kapcsolóival. Ha itt új adattípust választunk ki, akkor az a tárgy aktuális értékeivel létre is jön. Ha egy létező adattípus kapcsolóját deaktiválod, akkor az a típus törlődik a stateből. A state nevének megváltoztatása is lehetséges. Mindhárom műveletet azonban megelőzi egy biztonsági kérdés.

Most fantáziáljunk egy kicsit. Képzeld el egy ingát, amely ide-oda leng (*Azért ez merőben szokatlan egy ingától, nem igaz?*). Az előbb tanultak birtokában ennek elkészítése nem okozhat gondot, két state kell, amelyek a két végállapot Grouping információ-

## Detail editor

---

ját tartalmazzák. Most jön a design, az inga a két véghelyzetben legyen zöld, de a lengés során változtassa a színét úgy, hogy középre érve piros színben tündököljön. (*Ez ám a látvány!*)

A feladat nem is olyan bonyolult mint gondolnánk, középre kell állítani az ingát, színét átváltoztatni pirosra, majd ebből készíteni egy state-t, amelyben a csoportinformáció mellett az attribútumokat is tároljuk. A két végállapot statejeiben szintén tárolni kell a zöld színt. A lengés beállítása során az egyik végállapotból először a középső helyzetbe, majd onnan a másik végállapotba kell a tárgyat alakítani. Visszafelé ugyan így. A részlengések azonos számú képkocka alatt menjenek végbe. Ez eddig nem is okoz gondot, de hogyan állítsuk be pontosan középre az ingát? Erre van külön funkció, amely egy kiindulási állapotból megadott százaléknnyira alakítja át a tárgy tulajdonságait egy másik state felé. Ebben az esetben a két state között 50% átalakulás kell. A funkció végrehajtása a States kérdező **Tween** kapcsolójával kezdeményezhető. Működése során a jelenlegi állapotból, amely nem szükség szerűen definiált state, a listából kiválasztott state felé a szintén választható adattípusokat változtatja az input mezőben megadott százalékkal.

Van még két kapcsoló a States kérdezőjében, amelyről nem esett szó. (*Három, mert a Cancel-t sem magyaráztam meg részletesen.*) A **Clone** segítségével egy, már meglévő state-t klónozzhatunk. A klón az eredeti state mindazon adatait átveszi, amelyet az adattípus kérdezőben kiikszelünk. A klón a továbbiakban teljesen független az eredetitől.

Az **Import** segítségével a Detail editorban lévő valamelyik tárgy, vagy csoport jelenlegi állapotát importálhatjuk a kiválasztott tárgyba, megadott néven. Erre azonban van egy korlátozás, a két

tárgynak azonos struktúrájának kell lenni azaz azonos csoportszerkezettel kell rendelkeznie és azonos összetevő tárgyakból kell állnia.

### State trükkök

A száraz elmélet után üdítőleg hat néhány gyakorlati dolog a statekkel kapcsolatban. (*Egy üveg gyöngyözően hideg sör, amit a Bahamákban a pálmafák árnyékában egy fiatal szőke alig használt, tangában szolgál fel, na az hat üdítőleg, nem a hüjje statek!*) Az Action editorban, az animáció beállítása során közvetlenül csak a csoport összülőjének statejeihez férhetünk hozzá, ennek ellenére a csoport elemeinek lehetnek saját külön bejártatú statejei, amellet, hogy az összülő statejei belekerülnek a gyermekei statejeibe is. (*En speciel a tejberizst jobban szeretem, mint a tejbé state-t.*) Erről könnyen meggyőződhetsz, ha egy statekkel ellátott csoport valamely gyermekét választod ki, majd state infót kérsz róla. Más a helyzet, ha a kiválasztott tárgy azután került a csoportba, miután annak a statejei definiálva lettek. Ekkor nem kapja meg automatikusan a csoportra jellemző stateket, ezért nem is tud együttműködni azzal. Ez nem feltétlenül okoz hibát, sőt bizonyos esetben pont ez a lényeg, ezért előfordulhat, hogy a csoport egyes elemeiből utólag direkt törölünk definiált state-t, bár ez ritka.

Foglaljuk össze, amit az előző néhány zagyva sorban írtam:

A csoport elemei átveszik a felmenőikbe definiált state-t, ha annak létrehozásakor már a csoportban voltak. Egy gyermek object nem feltétlenül tartalmazza a szüleiben definiált nevű állapotokat. A csoport elemei rendelkezhetnek saját statekkel, amelyek nincsenek definiálva a felmenőiben.

## Detail editor

---

Az animáció szempontjából csak azok a statek jelentősek, amelyek az összülőben is meg vannak nevezve, mivel csak ehhez tudunk az Actionból hozzáférni.

Amikor az Action editorban kiválasztunk egy állapotot, annak neve "végigküldődik" a "családban" és minden tárgy, amelyben definiálva van ilyen nevű state, az annak megfelelően beáll. Amelyikben nincs ilyen state, az nem változik, kivéve, ha a szülője egy csoportszerkezet-változás miatt elmozdul, mert akkor vele együtt mozog az is.

Mire jó ez? Például arra, hogy egy bonyolult, sok tagból álló csoportot részekre bontva beállítunk, az egyes összetartozó állapotokat minden részben azonos névvel hozzuk létre, majd a végén az egészet újból összekapcsoljuk. Ezzel jelentős mennyiségű időt takaríthatunk meg, mivel a beállítás során kisebb méretű objecteket kell állandóan újrarajzolni a programnak, és a previewek generálása is kevesebb időt vesz igénybe. Az összekapcsoló tárgy lehet akár egy Axis is, ha előzőleg abban definiáljuk a stateket, de az egyik tetszőlegesen kiválasztott csoporthoz is hozzáköthetjük a többit. Az összekapcsoló tengelynek nem kell valós változást tárolnia, csak a nevek a lényegesek. Az sem fontos, hogy az egyes elemek ugyan olyan adattípusokat tároljanak.

Miután a stateket jól kielemeztük, ismerkedjünk meg más hasznos dolgokkal.

## Inverz kinematika

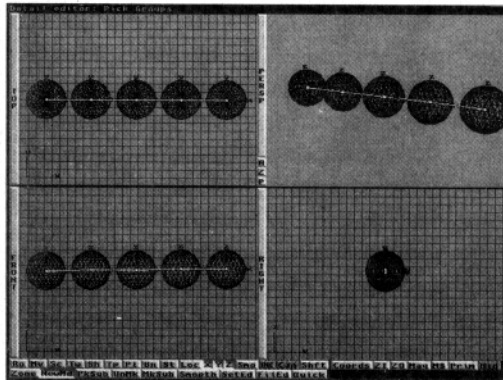
Rögtön az elején tisztázzuk mi az az inverz kinematika, illetve itt most mit értünk (*vagy nem értünk*) alatta. Tegyük fel, a valóságban felépítünk egy olyan tárgyat, mint egy object csoport. A csoport tagjait

kétféle módon kapcsolhatjuk egymáshoz: logikailag, amikor minden tárgy a másiktól függetlenül mozgatható, és fizikailag, amikor az elemeket fix hosszúságú, de esetleg elforgatható pálcák kötik össze. A valóságban persze ennél több lehetőség kínálkozik, de a program csak ezt a kettőt tudja modellezni.

Ezeddig csak az első, a logikai összekapcsolás modellezésével találkozhattunk, a másodikkal, a merev pálcás összekötéssel nem.

Ha a csoport elemeit merev pálcák kötik össze, az egyes elemek mozgása a pálcákon keresztül hatással van a csoport többi tagjára is. Ezt a hatást leíró fizikai törvényszerűség az inverz kinematika, amelyet a program is modellezni tud. A pálcák megfelelnek a csoport tagjait összekötő vonalaknak, a tárgyak pedig csuklópontként viselkednek. A programban minden csuklópontra külön meghatározhatjuk, hogy merre nem mozdulhat, vagy fordulhat el. Például ha a csoport egy emberi lábat modellez, akkor a térdizület helyén lévő csuklópont csak a kereszttengety körül fordulhat, de minden irányban mozoghat, feltéve ha a többi testrész engedi.

Legjobb lesz, ha ezt is gyakorlati példán szemléltetem. Készíts öt egyforma gömböt és kösd azokat sorba, ahogy a képen látszik. Ahhoz, hogy stabil tárgyunk legyen, amit pontosan lehet manipulálni, kell legalább egy fix pontjának lenni, ez





## Detail editor

---

az ősszülő, amelyet az inverz kinematikában mozdíthatatlannak, de elforgathatónak vesz a program.

Kapcsolj **Pick objects** módba és válaszd ki az utolsó gömböt. Próbáld elmozgatni. Semmi különös nem történik, a gömb mozog, mint ahogy azt márt megszokhattuk. Most pipáld ki a **States** menü **Constrain** pontját. Próbáld újból elmozgatni a gömböt. Na erről beszéltem, ez az inverz kinematika. Mindaddig, amíg a Constrain ki van pipálva, a tárgyak csoportbakötését jelző vonalak úgy viselkednek, mintha merevek lennének, rajtuk keresztül a mozgás átadódik a csoport többi elemére is. Fontos, hogy a kényszermozgatás során a tárgyak **Pick Objects** módban legyenek szelektálva.

Másik fontos dolog, hogy a kényszermozgatás csak interaktív manipulációkkal használható. A **Transformation** kérdezőben megadott elmozdulásoknak nem lesz hatása a csoport többi elemére.

Folytassuk tovább a kísérletezést, próbáljuk ki, hogyan lehet az egyes tárgyakat bizonyos mozgások ellen védeni. Ugyan csak Pick Objects módban válaszd ki a változatosság kedvéért az utolsó gömböt. A States menüből a **Freeze** ponttal hozd elő a korlátozások kérdezőjét. Rögzítsük a gömböt a Z és Y irányú mozgások ellen. Piciny angol tudással gondolom mindenki rájött, hogy a **Translation** jelenti a mozgás, a **Rotation** pedig a forgás korlátozását. De mit jelent a Local és a World? Azt, hogy az irányonként megadott korlátozás a tárgy saját, vagy a világ tengelyrendszere szerint értendő-e. Ezeket kombinálhatjuk is, azaz letiltható például a saját X tengely és a világ Z tengelyének irányába eső mozgás. A **Joints Lock** jelentését később ismertetem (*Fogadjunk nem is tudja, abban bízik, hogy addig valaki elmondja neki!*) Most csak a világ Z és Y

tengelyeit válaszd ki. Ezzel elértük, hogy a tárgy szabadon elfordulhat, de csak a világ X tengelyével párhuzamosan mozdulhat el.

Megjegyzem a Freeze-nek csak kipipált Constrain esetén van hatása.

Ha már a forgásnál tartunk, hol van a tárgy csuklópontja? Természetesen a tengelypontjában, mint ahogy ez látható, hiszen innen indulnak az összekötő pálcák.

Miután OK, válaszd ki most a *(Az utolsó gömböt?)* középső gömböt. Próbáld meg elmozgatni. Az utolsó gömb úgy mozog, mintha egy csúszdán lenne rögzítve. Ez természetes, hiszen ezt állítottuk be a Freeze-nél.

Az animációk során az előbb látott kényszermozgást szintén a statek segítségével lehet hasznosítani. Az inverz kinematika segítségével beállítjuk a tárgy kulcspozícióit, ezekről stateket készítünk a grouping paraméter típus eltárolásával, majd azokat az animációban az Action editornál leírt módon használjuk. A kulcspozíciókat jól át kell gondolni. Például egy teljes kör megtételéhez nem elegendő egy, de még kettő state sem. Ugyan is, ha van egy  $0^\circ$  és egy  $180^\circ$  elfordulást tartalmazó state, akkor még nem egyértelműen meghatározott, hogy a forgás mely irányban megy végbe. Ebben a konkrét esetben tehát három state kell, pl.  $120^\circ$ -os elfordulásokkal. Lehet az egyes elfordulások szöge különböző is, de ekkor nehezebb egyenletes forgást beállítani.

Az egyes tárgyak, illetve csuklópontok korlátozásait három féle módon oldhatjuk fel. A Freeze kérdezőben bármikor újból rendelkezhetünk az aktuális tiltásokról, vagy a **Release** ponttal az éppen kiválasztott csuklópont összes korlátozását törölhetjük. A **Release All** a kiválasztott tárgy összes csuklópontját felszabadítja.

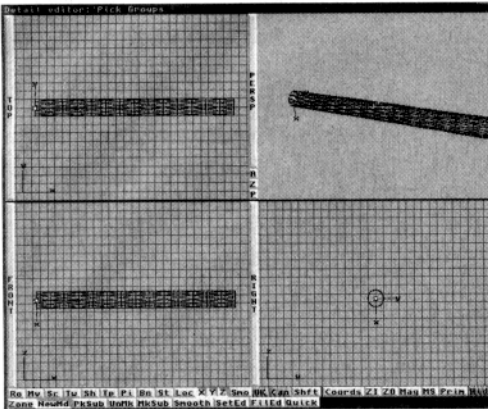
Közben kitaláltam mit is jelent a **Joint Lock**. (*Mondtam ugye!*) Ezzel az adott tengelyt a közvetlen felmenőjéhez kapcsolhatjuk, csak azzal együtt mozdulhat, fordulhat. Például ha Y-t Joint Lockoljuk, akkor az a tárgy Y tengelye mentén csak abban az esetben mozdulhat, ha vele együtt mozdul a felmenője is. Ha történetesen a felmenő ilyen mozgás ellen rögzítve van, akkor ez a tárgy is mozdíthatatlan lesz.

### Skeleton kontrol

Nem rossz dolog az inverz kinematika, de az még semmi, ahhoz képest ami most jön. Megismerkedünk a csontváz-as tárgymanipulációval. (*Tiszta horror, gyengébb idegzetűek fogják be a szemüket!*) A csontvázak lényege, hogy a tárgyhöz meghatározunk egy, csuklópontokkal összekapcsolt vonalakból álló, csontvázat. Ennek a váznak az alakítása azután átadódik a tárgyra és megfelelőképpen deformálja azt. A csontváz alakításához igénybe vehetjük az előző fejezetben ismertetett inverz kinematikát is. Gondolom nem kell nagy fantázia ahhoz, hogy elképzeld, milyen nagyszerű lehetőségeket rejt ez az animációs segédeszköz. Nagyobb grafikai munkaállomásokon már régóta használt és jól bevált fogás a csontváz-kontrol.

Az Imagine csontvázainak plusz képessége a személyi számítógépes platformon futó, csontváz-kontrollal rendelkező 3D programokéival szemben, hogy egyértelműen definiálható a csontváz által létrehozott deformáció helye, vagyis meghatározható, hogy a tárgyon hol az ízület.

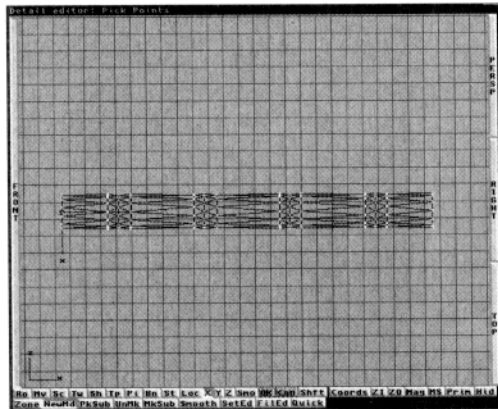
Nosza egy példa. Készíts egy primitív csövet, amely elegendő, ha 12 körszekcióból áll, de hosszában legyen 13 szegmenses. Méretét és helyzetét állítsd be



a képen látható módon. Lépj át Pick Points módba és rendezd át a tárgy pontjait, úgy, ahogy az a másik képen látható.

Az így létrehozott helyek lesznek a tárgy ízületei.

Most ehhez a tárgyhoz hozzunk létre egy csontvázat. Ez nem más, mint össze-groupolt tengelyek sorozata. A tengelyek jelképezik a csuklópontokat, míg az azokat összekötő vonalak maguk a csontok. Tegyel egy tengelyt minden ízület középebe, plusz egyet-egyét a tárgy két végére. Az egyik lesz a csont csatlakozópontja a képzeletbeli felsőbb tárgyhöz, a másikra csak azért van szükség, hogy a tárgyat a végénél tudjuk megfogni a deformáláshoz. A létrehozott tengelyeket sorban össze kell grupolni, mondjuk úgy, hogy a Tube tengelye felőli Axis legyen a csontváz összülője. A csontváznak nem szükséges a tárgyon belül tartózkodnia, mivel később úgyis meghatározzuk, mely csuklópont mely ízülethez tartozik.



## Detail editor

---

Ha idáig megvan, össze kel kötni egy csoportba a tárgyat és a hozzá tartozó csontvázat, mégpedig úgy, hogy a tárgy legyen a csoport őszülője. Ezzel még koránt sincs kész a dolog, hátra van a sokat emlegetett csuklópont-ízület egymáshoz rendelés.

Mielőtt ennek nekifognánk, tekintsük át az alapelvet. Az ízületet felületek határozzák meg. Minden ízület egy deformációs rész, a hozzá tartozó felületelemek deformálódnak, amikor az ízület hajlik. Az ízület "feletti" rész, azaz az ízülettől a tárgy tengelypontja felé terjedő felületekre az ízület nincs hatással, ellenben a másik irányban fekvő felületek deformáció nélkül elmozdulnak. Ennek igazolására figyelj meg pl. a mutató ujjad. Hajlítsd be a középső ujjpercnél. Az ujj csukló felé eső része nem változik (kivéve a bőr húzódását, de ettől most eltekintünk), az első ujjperc pedig elfordul. A kettő közé eső rész, az ízület eközben deformálódik, behajlik.

No, let's go! A csontváz minden tengelypontjához hozzá kell rendelni egy deformációs, és egy ezzel összefüggésben mozgó részt. Az előzőek során felületelekekről beszéltünk, tehát az összerendelést **Pick Faces** módban kell végrehajtani.

Két módszert használhatunk eme feladat megoldására, először az elsőt ismertetem. A kívánt felületelekekből alcsoportot kell létrehozni, majd azokat a megfelelő tengelyekhez hozzárendelni. Válaszd ki elsőnek a tárgy összes felületét és készíts belőlük egy alcsoportot a **Functions/Make/Make Subgroup** menüpont alkalmazásával, mondjuk "MIND" néven. Válaszd most ki az első ízület bal oldalától kezdve az össze felületet. Ez lesz az a rész, amelyre az adott ízületnek valamilyen módon hatása van, készíts belőle 1A néven subgroupot. Ezután ezen ízület jobb oldalától kezdve válaszd ki az összes fennmaradó részt, ez legyen az 1B alcsoport. Ezekre a felületekre

nincs deformációs hatással az izület, de annak hajlításakor ezek is elmozdulnak. Maga a deformálódó rész az 1A és 1B alcsoportok különbözete.

Ugyan ezen az elven kijelölt felületekből készítsd el a 2A, 2B, 3A, 3B, 4A, 4B alcsoportokat is. Az utolsó tengelyhez nem kell felületeket rendelni, mert ennek mindössze az a feladata, hogy a képzeletbeli ujjat a végénél tudjuk megfogni.

Az alcsoportok elkészítése után **Pick Objects** módban válaszd ki a csontváz legfelső tengelyét, majd a **States/Bones Subgrps** pontjának alkalmazásával hívd elő az összerendelés végrehajtását szolgáló kérdezőt. Itt a **Big Subgroup** input mezőben kell megadni annak az alcsoportnak a nevét, amelyre az adott csuklópontra hatása van, míg a **Small Subgroup** input mezőben azt, amelyet már nem deformál, csak mozgat. Mivel a legfelső tengelynél tartunk, mindkét input mezőbe a "MIND" csoportja kerül. Nem kell azonban a beírással manuálisan bajlódni, a **Browse** kapcsolóra klikkkelve előtűnik a definiált alcsoportok listája, amiből egy jól irányzott bal klikkkel választhatunk.

A következő tengelynél a Big Subgroup az 1A, a Small Subgroup pedig az 1B alcsoport lesz. Most jön a kreativitás, a másik három tengelyt állítsd be magad. Az utolsó tengelyhez nem kell alcsoportokat rendelni, mivel ennek nincs valódi izülete.

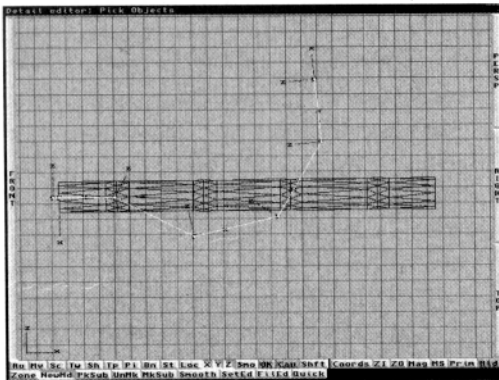
Hu, kész van, gyorsan mentsd ki! A hozzárendelések létrehozására van egy gyorsabb módszer is, amelyet az Imagine 3.1-es változatot használók élvezhetnek, a 3.0-ban ugyan is még nincs benne. Ez egy közvetlen lehetőség, nem kell alcsoportot képezni és külön a tengelyekhez rendelni. Az előző módon válaszd ki a megfelelő felületeket, ezután azokat a **Functions/Make/Mk. Big Subgrp**, vagy a **Mk. Sm. Subgrp** pontok alkalmazásával rögtön a

## Detail editor

tengelyekhez rendelheted. A menüpont használata után csak annyi a dolgod, hogy a kellő tengelyre kattintasz. Erre az állapot sorban egy figyelmeztetést is küld a program, amely mindaddig látható, amíg el nem találsz a tengelyt. Ha az utóbbi módszert használod, akkor is létrejön az alcsoport, ennek automatikusan adódó neve kerül be a Big és Small Subgroup input mezőbe.

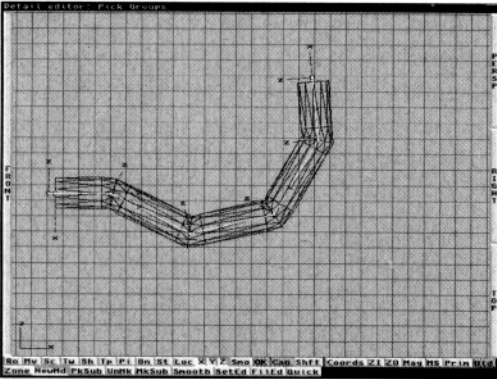
No, Setup OK, tovább. Nem elég az izületek meghatározása, kell egy rögzített alaphelyzet, amihez képest deformálhatjuk a tárgyat. Hol tároljuk el ezt az alaphelyzetet? Mondjuk egy state-ben, amelynek az egyértelmű megkülönböztetés végett adjuk a "DEFAULT" nevet. Ez a név ebben az esetben kötelező! A statesben most nem csak a group, hanem a shape információt is el kell tárolni. Előbbi a csontok helyzetének, utóbbi a tárgy alakjának rögzítése miatt fontos.

Most van kész, lehet örülni. Próbáljunk alakítani a tárgyon, ha már ennyit szenvedtünk vele. **Pick Objects** módban válaszd ki a csövet és a Freeze segítségével rögzítsd le mindenféle mozgás és forgás ellen. Ezzel azonos hatású, ha a csontváz összülő tengelyét rögzíted le, amely az egész tárgyat



képviseli. Ha ez a lerögzítés elmarad, a tárgy össze-vissza mozogni fog, miközben a csontvázat alakítjuk. Kapcsold be a **Constrain-t**, majd **Pick Objects** módban, ahogy az inverz kinematikánál láttuk, deformáld el a csontvázat. Ezzel

még nem deformálódik automatikusan a tárgy is, előbb ki kell választani azt, majd a **States/Bones Update** menüponttal alkalmazni a csont változását a tárgyra. Ez látszólag kényelmetlennek látszik, gyakorlatilag azonban nagyon hasznos, ha az emberfia nem egy erőműszerű munkaállomáson dolgozik, ugyan is egy összetettebb tárgy folyamatos



frissítése a csontműveletek során jelentős lassulást eredményezne. Ha meg vagy elégedve a tárgy új alakjával, ebből is készíts egy state-t, a többi már megy ahogy eddig.

Az előző néhány mondat úgy hangzik, mintha a csontváz

deformálása csak a Constrain bekapcsolásával, inverz kinematikai szabályok szerint mehetne végbe, holott ez nem így van. A Constrain kikapcsolása után a csontok szabadon manipulálhatók, akár még a hosszuk is megváltoztatható, a tárgy követni fogja ezeket a torzításokat is. A deformációs részek ebben az esetben is az ízületek lesznek.



### Tárgyalakító funkciók

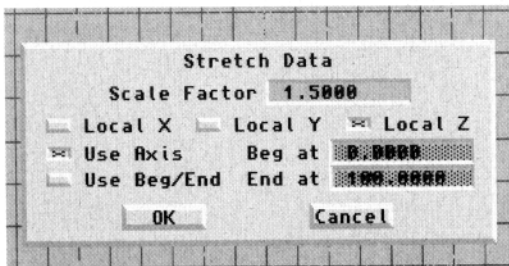
Kétség kívül az előző fejezetekben ismertetett lehetőségek a legfontosabb tárgyalakító műveletek, de ezzel nem merül ki a program tudománya. A tárgyak formálásához néhány új eszköz is az alkotó rendelkezésére áll. Ezeket a funkciókat interaktív és numerikus módon is igénybe lehet venni, csakúgy mint a korábban megszokott Move, Rotate és Size műveleteket.

A műveleteket a **Functions/Deformations** kérdezőjében találjuk. A listából valamelyiket kiválasztva megjelenik annak paraméterkérdezője, ahol a művelet adatait adhatjuk meg. Ugyan ezek interaktív változatai a képernyő alján lévő hat kapcsoló egyikével választhatók. A deformációk sokszor igen amorf dolgokat hoznak létre, ezért legjobban sok kísérletezéssel érthető meg a működésük.

A megfelelő forma eléréséhez a tárgynak elegendően sok pontból kell állni, például egy egy szegmensből álló csövet igen nehéz kihasasítani, hogy hordó alakú legyen.

A deformációk kérdezőjében sok azonos paraméter található, ezeket most összefoglalom.

A deformáció mértéke numerikus esetben a kérdező legtetején lévő input mezőben adható meg. Ez lehet méretarány vagy távolság, attól függően, hogy melyik funkciót használjuk.



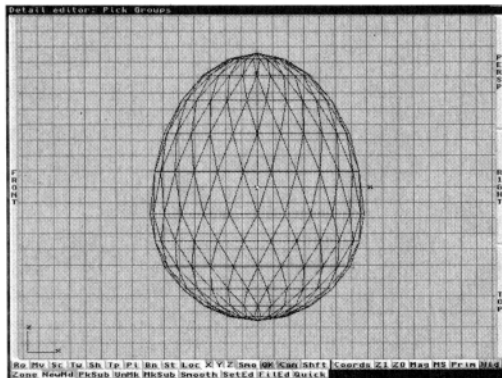
A deformáció valamely tengellyel párhuzamosan fejt ki hatását, ez a tengely mindig lokális, beállítani az adott művelet kérdezőjében a felső kapcsolósorral lehet.

Van úgy, hogy a művelet nem föltétlen terjed ki mind a két merőleges irányban, ekkor ezt az **X Only**, **Y Only**, és **X and Y** kapcsolókkal szabályozhatjuk a deformációk kérdezőjében. Ha nem a Z, hanem egy másik a méretváltozás tengelye, természetesen ezen kapcsolók jelentése is változik, és a fennmaradó két tengelyt kell rajtuk érteni.

A torzításokra jellemző, hogy nem föltétlenül a tárgy egészére, hanem akár csak egy ún. átmeneti részére fejtik ki hatásukat. Ez az átmeneti rész vagy a tárgy tengelypontjától a deformáció tengelyével párhuzamos pozitív irányba eső 100 egységnyi szakasz, vagy a tengelyponthoz képest megadott **Beg at** és **End at** közé eső rész. Hogy melyik, azt az **Use Axis**, vagy **Use Beg/End** kapcsolókkal választhatjuk ki. Megfelelően beállítva a paramétereket, a deformáció az egész tárgyra kiterjeszhető.

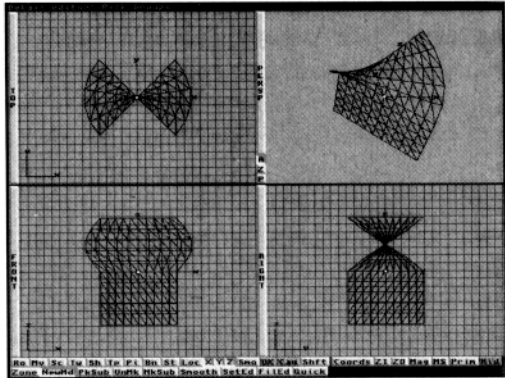
A következő műveletek közül választhatunk:

**Strech, St.** Ezzel egy tengely mentén tudjuk nyújtani a tárgyat, de nem lineárisan, mint a **Scale**-lel, hanem a nyújtás középpontjától távolodva egyre kisebb mértékben. Ennek megértésére próbáld ki a funkciót egy gömbön.



## Detail editor

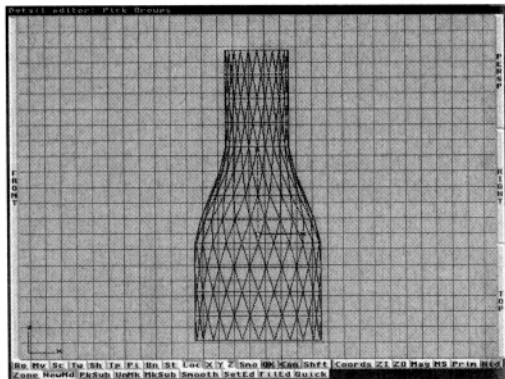
**Twist, Tw.** A tárgy egy részét lehet a funkcióval elcsavarni. A csavarás helye most is a tengely pozitív irányába eső 100 egységnyi rész, vagy a numerikusan meghatározott szakasz.



**Taper, Tp.** Szintén egy méretváltoztatás. A tárgy a tengelypontjától negatív irányban, vagy a numerikusan megadott rész kezdetéig nem változik, míg az átmeneti szakaszon egyre erőteljesebb a méretváltoztatás, a tárgy mintegy kihegyesedik. Az átmeneti szakaszt követő rész méretarány változása azonos mértékű.

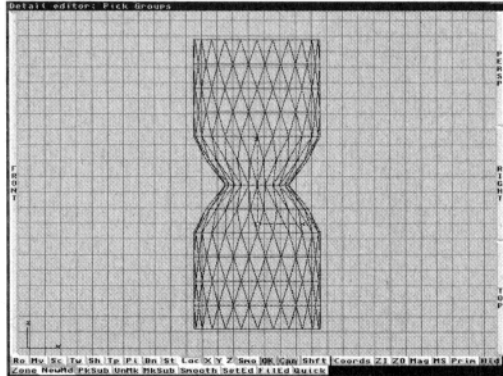
A **Smoot Taper** hatására a tárgy mértét változtató és nem változtató része lágyabban kapcsolódik egymáshoz, az átmeneti szakasz ívesebb lesz.

A funkció működése legjobban egy kb. 10-15 szegmensből álló tube primitíven szemléltethető. A primitív hossza legyen 200 egység. Ebből a Taper segítségével könnyen csinálhatunk egy tölcserőt. Ha

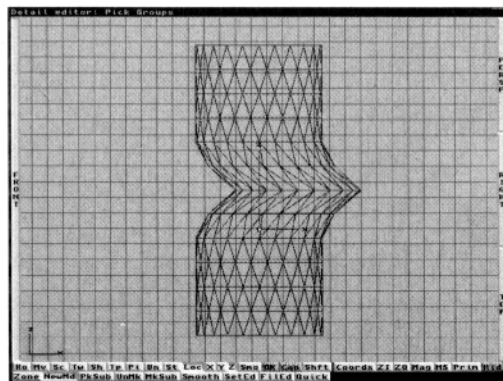


nem az X and Y van beikszelve, a tölcser lapított lesz, persze csak azon a részen amelyen a torzításnak hatása van.

**Pinch, Pi.** Nagyon hasonló a Taper-hoz, a különbség mindössze annyi, hogy az átmeneti szakasz után következő tárgy rész nem változtatja a méretét, a tárgy átmeneti részén mintegy behúzza, vagy kiereszti a hasát. A Smooth Pinch hatására most is ívesebb, simát átmenetet kapunk.

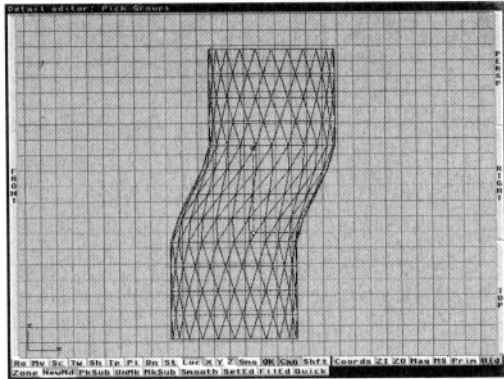


**Bend, Bn.** Ezzel a funkcióval a tárgy átmeneti részét a változás tengelyére merőlegesen valamelyik irányban eltolhatjuk, mint ahogy a hastáncosnő riszálja magát. Próbáld ki egy 200 egység hosszú, 12 szegmenses csövön. Azt, hogy melyik irányban riszálja magát a tárgy, az in X és in Y kapcsolókkal lehet megadni. Ha a torzítás tengelye nem Z, természetesen ez a két kapcsoló akkor is a fennmaradó irányokat jelenti. A **Smoth Bend** hatására a torzítás kevésbé lesz szögletes.



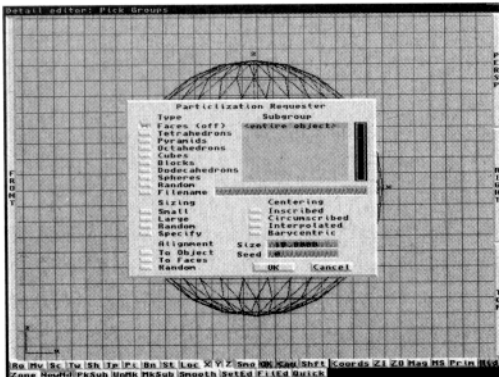
## Detail editor

**Shear, Sh.** Az utolsó torzító művelet, ez oldalirányban elnyomja a tárgyat, úgy, hogy a torzítás tengelyére merőleges élei továbbra is párhuzamosak maradnak egymással és az eredeti élekkel. Egy csövet így torzítva hasonló dolgot kapunk, mint az egymásra rakott, ferde aprópénzcsomó.



Van még néhány tárgyalakító funkciója a programnak, de ezeket nem a Deformationsban találjuk, hanem a **Functions** menüben önálló funkciókként. Legérdekesebb közülük a **Particles**, amellyel részecskékre bonthatjuk a tárgyat. Gyakorlatilag a tárgy minden egyes felületelemét lecseréli a program egy egyszerűbb tárgyra, például kockára. Ezek a részecskék nem záródnak teljesen össze, így az eredetileg folyamatos felületek szétszabdálódnak, de pont ez a lényeg.

Készíts egy primitív gömböt, válaszd ki és hívd meg ezt a funkciót. Megjelenik egy tekintélyes kérdező kismillió bizgentyűvel.



A **Subgroup** listában az **<entire object>** alatt az esetlegesen létező alcsoportok nevei láthatók. Itt választhatjuk ki, hogy a részecskékre alakítás a tárgy mely részeire terjedjen ki. Ezek kombinálhatók is, tehát egyszerre akár több alcsoportra is beállíthatunk különböző paramétereket.

A **Type** kapcsolóival lehet kiválasztani, hogy az érintett felületcsoport, vagy az **<entire object>** esetén az egész tárgy milyen formájú részecskékké alakuljon át. A **Faces (off)** az eredeti állapot, ekkor nem változik a tárgy, míg a többi kapcsoló a nevének megfelelő alakú részecskéket eredményez. Nem részletezem ezeket, csak kettőt emelek ki. A **Random** hatására a részecskék alakja véletlenszerűen választódik ki a fentebb lévő lehetőségek közül, míg a **Filename** opcióval egy megadott objectfájlban leírt alakot vesznek fel. A fájlt egy kérdezőn keresztül lehet megadni.

A **Sizing** kapcsolóival a részecskeméretet lehet szabályozni. A **Random** véletlen méretet határoz meg, a **Specify**-t választva pedig a **Size** input mezőben adhatjuk meg, egy-egy részecske méretét.

Az **Alignment** három kapcsolójával azt dönthetjük el, hogy a létrejövő részecskék a tárgyhöz, vagy az egyes felületekhez igazodjanak, esetleg véletlenszerűen váltakozva, hol ehhez, hol ahhoz.

Az utolsó kapcsolócsoport tagjaival szabályozhatjuk, hogy a részecskék a tárgy eredeti körvonalához képest hogyan helyezkedjenek el. Az **Inscribed** belülírot jelent, vagyis a részecskék az eredeti alakon belül helyezkednek el. Ennek ellentettje a **Circumscribed**, a körülíró, amikor a részecskék az eredeti alakot kívülről érintik. Az **Interpolated** azt jelenti, hogy a részecskék középvonala kerül az eredeti alakra.

## Detail editor

---

A **Seed** nevű input mezőben egy tetszőleges számot adhatunk meg, ez lesz a véletlen szám generátor kezdőértéke.

A **Particlization** funkciónak van egy hibája, a Detail editorban nem követhető nyomon a hatása, a részecskék csak a renderelt képen látszanak. A szerkesztőkben a tárgy mindvégig az eredeti alakjában szerepel.

A részecskékre bontáshoz hasonló hatású az **Object/Fracture** funkció, amely nem csak az élek felezésére használható, hanem a tárgy felületekre bontására is. Készíts egy gömböt, válaszd ki, majd alkalmazd rá a **Fracture**-t. Megjelenik egy kérdező, amely a **Scale Factor** felől érdeklődik. adj meg mondjuk 1.5-öt. Az enter lenyomása után a gömb felületeire esik szét, amely felületek ráadásul még el is távolodtak egymástól. Az eltávolodás mértéke az előbb megadott számtól függ. Ha a **Scale Factor** 1.0, a szétbomlott felületek mindegyike eredeti helyén marad, a tárgy látszólag nem változik. Ez azonban csak látszat, valójában sokkal több pont és él fogja alkotni. Ha a **Fracture** hatásának kitenni szándékozott tárgyat tartalmazott, egy figyelmeztetés jelenik meg, miszerint a művelet hatására a statek adatai elvesznek.

Következő funkció, amellyel a tárgy alakja változtatható, az **Applyque**. Ezzel egy **IFF** (a PC-s változatban **TIFF**, **TARGA**, **PCX**) képpel torzíthatjuk el a tárgyat, hasonló módon, mintha azt a képet altitude mapként feszítettük volna rá. Lényeges különbség azonban, hogy ez a funkció nem csak látszólagosan, hanem valójában és véglegesen átrendezi az alkotó pontokat. Az egyenetlenségeket kialakító képre a funkció végrehajtása után már nincs is szükség.

A **Weave** segítségével meghullámoztathatjuk a tárgyat. A hullámok a tárgy saját **XY** síkján jönnek létre, a hullám amplitúdója **Z** irányú. A hullám középpontja egybeesik a tárgy tengelypontjával, legnagyobb amplitúdóját a tárgy **Z** tengelyének mérete határozza meg. A hullámok erőssége a tengelyponttól távolodva csökken, az **X** és **Y** által mutatott távolsághoz érve értékük nulla.

Készítsünk most egy piciny animatát, amelyben egy sík lapot hullámoztatunk. Először is hozz létre egy primitív plane-t, de 20X20 pontból. Minél több pontból áll a tárgy, annál finomabb lesz a hullám-mozgás, mivel ez az alkotó pontok átrendezésével manipulál. Forgasd el a tengelyét **X** körül 90 fokkal. Erre azért van szükség, hogy a **Z** tengely a felületre merőlegesen álljon. A sík ezennel elő van készítve, használd a Wave funkciót. A létrejött alakból készíts egy state-t, a shape információ eltávolításával. Lépj be a transformation kérdezőbe és a Size-t választva írd be **Z**-hez -1-et. Ennek hatására a tárgy alakja az **XY** síkra tükröződik, a hullám megfordul. Ez az alak is megér egy state-t. Állítsd be a perspektíva nézetet úgy, hogy a síkot éléről lásd, majd készíts egy state animot, amiben az alak egyik állapotból átmegy a másikba és vissza.

Most pedig ablakozzunk. A **Latticize** funkció segítségével a tárgy minden felületelemére ablakokat vághatunk, ezzel olyan hatást kiváltva, mintha a tárgynak csak a váza volna meg, de valójában nem tartalmazna felületeket. Készíts egy plane-t, de elég ha 4X4 elemből áll. Válaszd ki a tárgyat és alkalmazd a Latticize-t. Megjelenik egy input mező, amelyben azt adhatjuk meg, hogy az eredeti élek helyén a felületelemhez képest milyen széles ablakkeretek jöjjenek létre. Legyen ez most 0.1, vagyis minden felület 90%-a helyén lyuk lesz, a maradék



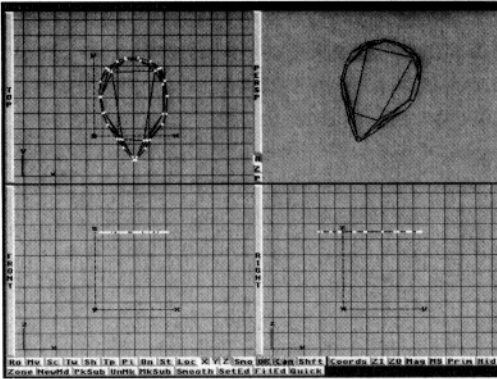
## Detail editor

---

10% pedig keretezi ezeket a lyukakat. A hatást lemérheted, ha a perspektívaablakot kinagyítod és a Shaded megjelenítést választod.

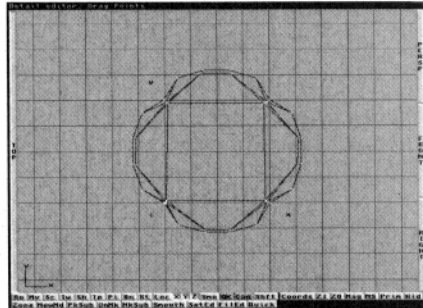
Most pedig bemutatok egy érdekes és nagyon hasznos tárgyalakító funkciót. Bonyolult, sok pontból álló tárgyak alakítása sokszor igen nehézkes, hiszen minden pontot megfelelő arányban kell mozdítani. Addig nincs is baj, amíg a Deformations elemeivel megoldható a torzítás, de mit tegyünk például akkor, ha egy hengeres korsó szájára szeretnénk kiöntőt alakítani. A Forms editorban könnyen megoldható a dolog, de ha egy Detail objecten szeretnénk ezt megvalósítani, amely nem vihető át a Forms-ba, már nehezebb a helyzet. Ilyen és hasonló feladatok megoldására fejlesztették ki a **Deform Tool** nevű eszközt az Imagine alkotói. A Deform Tools egy tetszőleges méretű szegmensszámú téglatest, amely úgy viselkedik mint egy felületek nélküli, csak pontokból és élkből álló tárgy. Ez azonban "emlékszik" kezdeti alakjára és nyilvántartja, hogy ahhoz képest milyen deformációkat hajtottunk végre rajta. Ha a megfelelő alakítás után a belsejébe teszünk egy normál tárgyat, kívánságunkra a Deform Tool alakításai átadódnak erre a tárgyra. A DT-nek és az alakítandó tárgynak nem kell azonos felépítésűnek lenni, azoknak a pontoknak, amelyek nem esnek egybe a két tárgyon, interpolációval számolódik ki az új helyzete.

Nézzünk egy konkrét példát a homály oszlatásához, próbáljuk megcsinálni a fentebb említett korsót. Ezt majd egy alap Tube primitívből képezzük, de előbb készítsük el a deformáló eszközt az **Objects/Add/Deform Tool** pontjával. Mint említettem a DT egy téglatest, méretét és szegmenseinek számát mindhárom irányban külön megadhatjuk a megjelenő kérdezőben. A méret most legyen



rendezd át a pontokat, hogy a képen látható alakzatot kapd.

A felülnézetben alul látható kicsúcsosodás fogja létrehozni a kancsó kiöntőjét. A hatás szemléletesebbé tétele érdekében készítsünk egy talpat is a korszónak, takard most ki a felső két pontsorát a DT-nek, az alsó pontsorból pedig hozd létre a következő alakot:



Lépj vissza Pick Groups-ba és készítsd el a csövet, amely majdnem egy alap paraméterekkel rendelkező Tube primitív, egyetlen különbség, hogy két függőleges szekcióból álljon. Vedd egy kicsit kisebbre scale-lel, mivel az alakítandó tárgynak benne kell lenni a DT eredeti mérete által határolt területrészek. Ha kisebbre vetted, állítsd be a pozícióját is, hogy a fenti feltételnek megfeleljen. Ezután, miközben a henger van kiválasztva, alkalmazd rá a **Functions/Deform Tool** funkcióját. Megjelenik egy kérdező, amelyből a deformáló

## Detail editor

---

eszközt kell kiválasztani, legyen ez most az imént létrehozott DTOOL\_5\_5\_2. A program szöszmötől egy kicsit, mialatt a DT deformációit átülteti a tárgyra. Ha jól követted a fent leírtakat, kb. ilyen alakot kapsz:



Ugye könnyebb így? Persze nem csak egy-egy szint alakja módosítható egy időben, hanem akár az egész tárgy is. Minden pont a DT eredeti alakjában hozzá legközelebb eső két pontot összekötő vonal elmozdulása szerint változtatja a helyét.

A tárgyakat nem csak a beépített alakító funkciók segítségével alakíthatjuk, hanem hagyományos módon, az alkotó pontjaik szerkesztésével is. Ekkor azonban szoszor problémát jelent a takarásban lévő pontokhoz való hozzáférés. A program korábbi vál-

tozatában erre a problémára már láttunk egy megoldást, a Hide funkció személyében, mellyel kitakarhatjuk a felesleges pontokat. Ezt a lehetőséget a program új változatában jelentősen kibővítették. Ahhoz, hogy a kitakart pontokat újból láthatóvá tegyük, eddig vissza kellett lépni Pick Groups, vagy Pick Objects módba, majd a Hide Pointsba ismét belépve a feleslegeseket újból kitakarni. Ezt feladatot leegyszerűsíthetjük a **Pick/Select** menü **Unhide All** pontjával, amely függetlenül attól, hogy melyik szerkesztési módban vagyunk, ismét láthatóvá teszi az elrejtett pontokat. A másik hasonló funkció szintén láthatóvá teszi az elrejtett pontokat, de nem az összeset, hanem csak egy megadott subgrouphoz tartozókat. Miután a **Pick/Select** menüből kiválasztottuk az **Unhide Subgroup** menüpontot, egy subgroupot kérdező jelenik meg. A kiválasztott alcsoport pontjai ezután láthatóvá válnak.

A Hide módban történő pontrejtegetésnek hátránya, hogy az elrejtendő pontokat kell benne kiválasztani, holott általában egyszerűbb volna a megmaradóakat jelölni. Pontosán ezt a célt szolgálja **Pic/Select** menü **Hide Unselected** pontja, hatására csak a szelektált pontok maradnak láthatóak, a ki nem választott pontok elrejtődnek. Ezt nem csak Pic Points módban, hanem Pic Edges és Pick Faces-ben is használhatjuk, ekkor a kijelölt élékhez, vagy felületekhez tartozó pontok maradnak láthatóak.

Most pedig adok egy házi feladatot. Készíts egy félgömböt, de ez legyen teljesen zárt, azaz a metszeti részen is legyen felülete. A gömböt félbevágni igen egyszerű, Pick Points módban ki kell választani a felesleges pontokat és törölni azokat. Mivel így a metszeti részen nem lesz felület, ez még csak a feladat fele. Eddigi ismereteink szerint tudunk erre többféle megoldást is találni. Az egyik, hogy a

## Detail editor

---

metszet peremén lévő pontok kivételével minden pontot elrejtünk, majd a megmaradók közé manuálisan felületeket képezünk Add Faces módban. Lassú, nem pontos, fáradalmas. Másik megoldás, hogy lemásoljuk a tárgyat, a másolat metszetének peremét levágjuk, térben kinyomjuk, majd a Slice funkció segítségével kivágunk vele egy síklapból egy kört, amely pont illeszkedik a félgömb nyitott felületére, összemergelve vele megoldottuk a feladatot. Még lassabb, pontos ugyan de fáradalmas. (*Lustaság fél egészség, a kosz meg melegít*) A Imagine 3.2-ben van erre egy igen elegáns megoldás. **Pick Edges** módban válaszd ki a félgömb peremén lévő élek felét félkörben, majd hívd meg a **Functions/Set Edge Line** menüpontot. Ezzel a művelettel a kiválasztott éleket célként kijelöltük egy felületképző eljárás számára. Most válaszd ki a perem fennmaradó éleit és hívd meg a **Functions/Fill to Edge Line** menüpontot. Ez a szelektált éleket felületekkel kapcsolja össze az előzőleg kijelölt élekkel. A funkció működése hasonló a Skin-hez, de egyszerre mindig csak két élcsoport között működik, viszont nem feltétele, hogy ezek azonos számú és sorrendű pontból álljanak. Nem csak ilyen egyszerű feladatokra használható, hanem például két tárgy egybefűzésére is. Tegyük fel, egy 24 szegmensből álló hengert kell kapcsolni egy 16 szegmenses kúphoz, hogy azok zárt felülettel illeszkedjenek. A Join és Merge használatával ez igen nehéz, nem kapunk sima illeszkedést a két tárgy között. Nézzük, hogyan oldható meg a feladat az előzőleg megismert funkciópárossal. Természetesen a két tárgyat most is egybe kell kapcsolni a Joinnal, vagy a Merge-dzsel, de úgy, hogy maradjon közöttük egy szűk sáv, ahová a kapcsoló felületeket létrehozuk. Ha ez

megtörtént, a kiválasztott tárggyal lépj át **Pick Edges**be. Válszd ki a hengernek a kúp alsó része felé eső peremenek éleit és állítsd be a kapcsolódás céljának a Set Edge Line-nal. Ezután a kúp peremén lévő élek szelektálása következik, majd mindennek tetőzéseként a Fill to Edge Line, amely létrehozza a kívánt felületeket. A funkció tehát, mint láttuk, két bármilyen bonyolultságú tárgy felületekkel történő egybefűzését képes elvégezni, gyorsan, pontosan, kényelmesen.

Következő feladat. Készítsünk egy hőlégballon alakú tárgyat. Hohó, mi sem egyszerűbb ennél, körvonalat megrajzoljuk, középpont körül megforgatjuk, kész. Ahogy a Möricka elképzeli. Az első problémák már a szép ívelt körvonal megrajzolásakor jelentkeznek. Seba, Drag Points-ba átváltva röpké negyed óra alatt sikerül átrendezni a pontokat, hogy a megfelelő körvonalat többé-kevésbé megközelítsük, az a kis szögletesség, ami marad, az még megengedhető. No, erre is tudok egy jobb megoldást, illetve nem is én, hanem az Imagine. Rajzold fel Add Linesben a körvonalat durva közelítésben, majd Pick Edgesbe átlépve válaszd ki azokat. Fontos, hogy az élek összefüggő vonalat alkossanak. Most jön a csel, a **Functions/Smoot Edge Line** képében. Hatására megjelenik egy input mező, benne egy 3-as. Ezzel az értékkel tudjuk a görbe lesimításának mértékét szabályozni. Hogyan? Hm, nos erre a fogós, ravasz kérdésre maguk a programozók sem igazán tudják a választ, lényeg, hogy ez egy polinominális érték (*nem mondanám, ha nem ez volna a kézikönyvben írva*), amely szabályozza a funkciót. Próbáld meg vele, többet a szerzők sem tudtak mondani. A lényeg, hogy az Enter lenyomása után a kiválasztott élek úgy rendeződnek át, hogy a görbe lágy lefutású lesz. Ha a polinominális érték nem megfelelő, egy hibaüzenet

## Detail editor

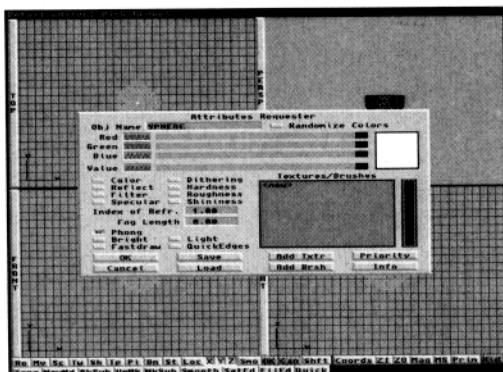
lesz a válasz, más nem történik, próbálkozz más polinominális értékkel (tudom hogy ez szóismétlés, de olyan szép az a polinominális érték, hogy muszáj volt még egyszer leírni). Szóval polinominális érték, ne feledd!

### Attribútumok

A tárgyak attribútuma nem lényegében változtak meg, de az attribútum kérdező tartalmaz egy újdon-ságot. A tárgyra immár tetszőleges számú textúrát és brusht helyezhetünk, egymással keverve is. A prioritás szempontjából tehát nincs különbség a textúra és a brush között, azt a listában elfoglalt helyük határozza meg. Textúrát/brusht a tárgyra helyezni az **Add Txtr** és **Add Brsh** kapcsolókkal lehet, a korábbi

programváltozatok-kal azonos módon. A prioritási sorrend kezdetben megegyezik a definiálásuk sorrendjével, az új textúra/brush mindig a kiválasztott mintázat elé kerül, ha ilyen nincs, akkor a lista végére. A sorrend megváltoztatását szolgálja a **Priority** kapcsoló. Ezzel az éppen kijelölt mintázat prioritását adhatjuk meg, az enter lenyomása után az adott mintázat elfoglalja az új helyét.

A már elhelyezett mintázat kérdezőjébe úgy léphetünk be, hogy a módosítani kívánt minta nevét kiválasztjuk a listából, majd az Info gombra klikkelünk.

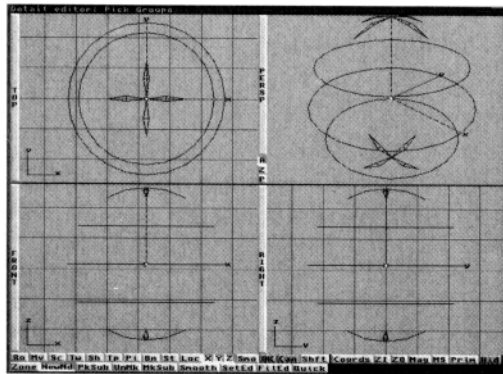


A textúrákkal annyi változtatás történt, hogy a régiek nem használhatók, az Imagine 3.0-tól kezdve más a textúra fájl formátum. Ez fordítva is érvényes, az új textúrák sem használhatók a régi programmal. Ráadásul ez a változat már figyeli a textúra fájlok kiterjesztését, és csak a \*.itx fájlok kerülnek be a kérdezőben a listába, ahol a nevük a kiterjesztés nélkül jelenik meg.

Az új textúrák nagy része kooprocesszort igénylő kóddal van írva, a jelentős számításiigényességük miatt, ezért a nem kooprocesszoros program változatot használók ne lepődjenek meg, ha nem minden textúrát tudnak használni. Az új textúrák részletes leírása külön fejezetben található.

A brushoknál is történt néhány apróbb változás. Megjelent egy új brush típus, a **Reflection**. Ez nem a tárgy valamely attribútumát változtatja meg, mint például a Reflectivity, amely a fényvisszaverő képességet módosítja, hanem az így felhelyezett kép úgy jelenik meg, mintha az a tárgyon tükröződne. Fraktál zaj típusú képek ilyenén alkalmazásával igen élet-szerű fém felületek hozhatók létre, gyönyörű csúcsfényekkel. Felhasználható még ez a funkció nem trace módban készült képeknél a valódi tükröződések utánzására.

A tárgyak az új program változatban egy attribútummal többet mondhatnak magukénak, ez pedig **QuickEdges**, amely a FastDraw-hoz hasonlóan egy gyorsított megjelenítési eljárás. A különbség, hogy nem



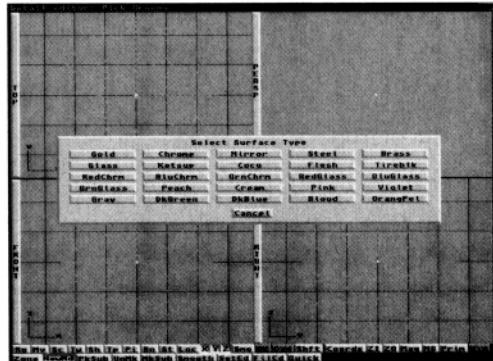


## Detail editor

egy befoglaló kerettel imitálja a tárgyat, hanem mindössze csak bizonyos éleit mutatja. Ezzel összetett alakzatok esetén is fogalmunk lehet a tárgy valódi alakjáról, amellet, hogy élvezhetjük a gyorsított megjelenítés előnyeit.

Azokat az éleket, amelyeket a program a **Quick Edges** attribútumú tárgyak megjelenítésére használ, előre definiálni kell. Ezt **Pick Edges** módban tehetjük meg, a kiválasztott él **Functions/Make/Make QuickEdges** funkcióval történő megjelölésével. A tárgy alakítása során előfordulhat, hogy valamely él, vagy élék, amelyek eddig **Quick Edges**-nek voltak felhasználva, már nem jellemzőek a tárgy alakjára, kívánatos volna ebbéli szerepüket törölni. Erre szolgál az **Unmake Quick Edges** funkció, amely a kijelölt él **QE** attribútumát visszavonja. Hasonlóan az alsoport tagsághoz, ez sem módosítja a tárgy alakját, a **Quick Edges** is csak egy jelzés. Ide kapcsolódik, hogy a **Functions** menü **QuickDraw Toggle** menüpontjának hatására előbb egy kérdező jelenik meg, ahol ki kell választani, hogy a hagyományos, vagy a **QD**-s megjelenítést akarjuk-e kapcsolni, csak azután hajtódik végre a funkció

Van egy új attribútum-hozzárendelési lehetőség is, a **Quick Attributes**-ek által. Ezek azonosak a normál attribútumokkal, mindössze a tárgyhoz rendelésük módja más. A leggyakrabban használt 25 attribútum csokorba fogható és ezek közül akarmelyik egy külön kérdezőn keresztül egyszerűen a tárgyhoz rendelhető.



Nézzük hogyan. Válassz ki egy tárgyat (lehető többet is egyszerre), majd a **Functions** menü utolsó pontját aktiváld. Megjelenik egy 5X5 kapcsolót tartalmazó ablak, ahol az egyes kapcsolók jelentik az attribútumokat. Ha ráklickepsz az egyikre, az annak megfelelő attribútum hozzárendelődik a tárgyhoz, mintha az attributes kérdezőben a Load-dal betöltöttük volna. Látható, hogy így lényegesen egyszerűbb.

A kapcsolók feliratait és a hozzájuk tartozó attribútumok megváltoztathatók. Mégis hogyan? A 25 attribútum a program könyvtárából nyíló **Attribs** könyvtárban helyezkedik el **X.atr** néven, ahol X az attribútum sorszáma 1-25 között. Az attribútumok azonos szerkezetűek azokkal, amelyeket az Attributes kérdezőből mentünk ki, létrehozni úgy lehet őket, hogy egy axist ellátunk a megfelelő beállításokkal, majd kimentjük azt az Attributes kérdező Save pontjával a megfelelő néven.

A kérdezőben a kapcsolók számozása balról jobbra és fentről lefelé halad. Például az 1.atr a bal felső, a 6.atr az alatta lévő, stb.

A kapcsolók neveit a **Preferences** editorban állíthatjuk be az **ATxx** sorokban. A nevek legfeljebb tíz karakterből állhatnak. Az ATPT sorban az attribútumok alapkönyvtárának nevét definiálhatjuk.

Ugyan itt, az **ATPT** sorban adhatjuk meg annak a könyvtárnak a nevét, amelyben a quick attribútumok vannak tárolva. Ennek a könyvtárnak mindenképpen a program könyvtárából kell nylni.

A quick attribútumok alapkönyvtárát az **ATPT** sorban állíthatjuk be.

### Fényforrások

A fényforrások paraméterei is bővültek, próbából hozz létre egy tárgyat, majd az attributes kérdezőjében kapcsold be a **Light** kapcsolót. Megjelenik a **Light Source Data** kérdező, amelyet a Stage, vagy Action editorban Light Source színrevitelekor láthatunk, ezért jól figyelj, mert ott nem mondom el még egyszer.

Az **R/G/B Intensity** gondolom nem okoz gondot, ez a fényforrás színe és ereje. Emlékezz rá, mit írtam az előző kötetben, az egyes színösszetevők értéke maximum **32768** lehet, bár ekkor már atomvillanás erejű a fény.

A fényforrás, a belőle áradó fény szempontjából kétféle lehet, pontszerű, azaz a fénye egy pontból ered, vagy párhuzamos sugarú, amikor is a fény a fényforrás X-Z tengelye által meghatározott térrészből ered, és párhuzamos sugarakat vetít Y irányába.

Nézzük az előző esetet, a fény egy pontból, a fényforrás tengelypontjából ered. Ezt a Point Source



kapcsoló kiikszelése eredményezi. Ha másként nem rendelkezünk, a fény a tér minden irányában egyforma mértékben terjed. Másként rendelkezni a **Round/Rectangular Shape** kapcsolókkal lehet, ekkor a fény irányítottá válik, csóvában terjed Y tengelye mentén. Előbbi hatására a csóva alakja hengeres, a fénykör sugara a tengelyponttól Y távolságra X lesz, ami megfelel a régi program által kínált irányított fényű fényforrásnak. Utóbbi kapcsolót aktiválva a fénycsóva keresztmetszete négyzetes lesz, a fényfolt alakja Y távolságban megfelel az X-Z tengelyeinek.



Ha a fény párhuzamos sugarú, akkor mindössze annyi a különbség, hogy a fénycsóva nem kúp, vagy gúla alakú lesz, hanem henger, vagy téglatest, amelyben a fénysugarak a fényforrás Y tengelyével párhuzamosan terjednek.

## Detail editor

---

A **Cast Shadow** nem változott, aktiválásával érhetjük el, hogy az adott fényforrás vessen árnyékot trace képszámítási módban. Ha ez inaktív, a fényforrás semmilyen körülmények között nem vet árnyékot.

A **Diminish Intensity** szintén a régi, ha aktiváljuk, a fény a távolsággal arányosan csökkenő intenzitású lesz. Hogy mekkora lesz az intenzitáscsökkenés, a fényforrás Y tengelyének mérete dönti el, minél nagyobb, annál nagyobb távolságot tesz meg a fény, mire fele intenzításra csökken.

A **Controlled Falloff** új dolog, ezzel szintén csökkenő intenzitásúvá tehetjük a fényt, de keresztirányban, azaz a fénycsóva a széle felé gyengébb fényt fog sugározni. Az intenzitáscsökkenést most is a fényforrás Y tengelyének mérete szabályozza.

A **No Lens Flare** kapcsolóval azt tudjuk beállítani, hogy a LensFlare, vagy NewLens global effect használata esetén ez a fényforrás okozzon-e kameracsillanást, vagy sem. Ha aktiváljuk, az adott fényforrás nem okoz Lens Flare-t.



## Egyéb újdonságok

Ebben az alfejezetben említem azokat az új, vagy megváltozott funkciókat, amelyekről eddig nem esett szó, mert nem illettek be a tanrendbe. Az egyszerűség és az áttekinthetőség miatt a funkciók többé-kevésbé a menükben elfoglalt sorrendjükben lesznek megemlítve, ettől csak az egymással kapcsolatos, de más helyen lévő dolgok esetén tesztek kivételt.

Az **Object** menüben találjuk a **Check Obj** funkciót, amely a kiválasztott tárgy logikai helyességét ellenőrzi és tájékoztat az esetleges hibákról. Az egyik leg-sűrűbben előforduló üzenete, hogy a két pont teljesen azonos pozícióban van. Ez nem föltétlen hiba, néha kifejezetten ez a cél, de legtöbbször fölösleges, és növeli a fájl méretét. A duplázott pontokat nagyon egyszerűen el lehet tüntetni. Válaszd ki a tárgyat, majd alkalmazd rá a Merge-öt. Ez a funkció eredetileg több tárgy optimalizált egymásba fűzését szolgálja, de a tárgyon belüli duplikált pontokat is kiszűri és megszünteti.

A **Pick/Select** menü **Pick More** funkcióját **Pick Faces** módban használva, azokat a felületelemeket is hozzávehetjük a kijelöltek csoportjához, amelyek közvetlenül csatlakoznak azokhoz, vagyis van az épen kiválasztott felületekkel közös pontjuk. No, ez így kissé zavaros. Készíts egy gömböt, majd lépj át vele Pick Faces-be. Válassz ki rajta középtájon egy felületelemet (*még véletlenül sem a gömb csúcsánál, vagy valamelyik sarkán!*). Most használd a Pic More-t, szóval ez az.

Utóbbi funkció nem csak **Pick Faces**, hanem **Pick Edges** és **Pick Points** módban is használható (*Pick Szalámiban nem*). Pick Edges-ben azon felületek határoló éleit, Pick Points-ban pedig azok határoló pontjait teszi szintén kiválasztottá, amelyeknek van közös pontja a már kiválasztott élekkel/pontokkal.

## Detail editor

Elérkeztünk a Detail editor utolsó újdonságához, ez pedig a **Pick/Select** menüben lévő **Edge Filter**. A kiválasztó funkciót **Pick Edges** módban tudjuk használni bizonyos élek kiválasztására.

A funkció aktiválása után megjelenik egy kérdező, amelyben a kiválasztandó élek feltételeit lehet beállítani. Ha az **Use Angles** kapcsolót kiikkszeled, a **Min** és **Max** input mezőben megadott szögértékek közé eső élek kerülnek kiválasztásra, vagy vesztek el kiválasztottságukat.

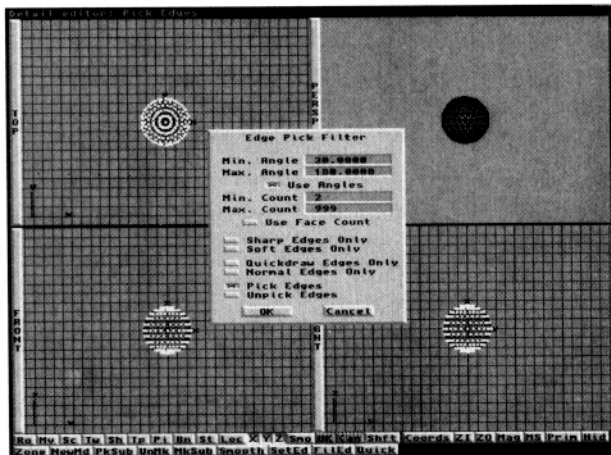
Az **Use Face Count** kiikkszelése után azok az élek kerülnek a funkció hatása alá, amelyek a **Min** és **Max** közötti számú felülethez tartoznak.

A **Sharp/Soft Edges Only** kapcsolókkal szabályozhatjuk, hogy a művelet csak a soft, vagy csak a hard élekre legyen hatással.

**Quickdraw/Normal Edges Only** hasonló, mint az előző, de csak a **Quickdraw**-ban résztvevő, vagy részt nem vevő élekre lesz hatással a funkció.

A fenti lehetőségek kombinálhatók, több feltétel támasztása esetén csak azok az élek befolyásolódnak, amelyek minden feltételnek megfelelnek.

Az **Edge Filter** terrel nem csak kiválasztani, de ki nem választottá is lehet tenni az érintett éleket, hogy melyiket akarjuk, az a **Pick/Unpick Edges** kapcsolókkal dönthető el.

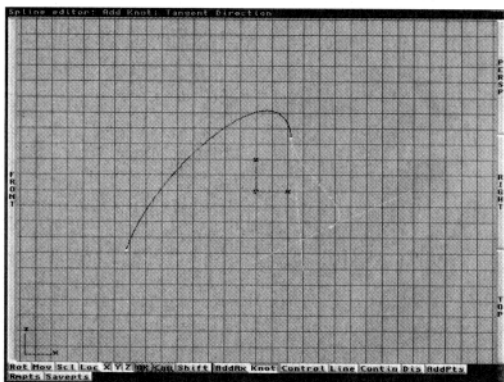


# Spline editor

A Spline editor egy vadonatúj szerkesztő az Imagine-ben, segítségével spline, azaz görbe vonal határolta tárgyakat szerkeszthetünk. Ezek legfőbb előnye a poligonálisan szerkesztett tárgyakkal szemben, hogy a görbe vonalak pontosabban, szögleteség nélkül megrajzolhatók. Mivel a program többi része azonban poligonális, ezért a rendering, vagy a Detailban történő további alakítás előtt át kell alakítani a tárgyakat poligonális formába. A spline alapú szerkesztés előnyeit azonban ekkor is élvezhetjük, mert a görbék poligonná történő konvertálása igen pontos, jó felbontású, sokkal jobb eredményt érhetünk el, mintha eleve poligonálisan hoztuk volna létre azt.

A másik jellemző felhasználási területe a Spline editornak a font objectek szerkesztése. Segítségével **Adobe Type 1**-es formátumú fontokat kezelhetünk, amelyekből tetszőleges feliratot ábrázoló objectet hozhatunk létre. A program ezekből térbeli alakzatot állít elő, és képes az extrudálás során az élek letörésére, gömbölyűbb kinézetű tárgyak létrehozására. A font objectek képzéséhez a **\*.pfb** kiterjesztésű fájlok kelljenek.

A görbe csomópontokon, "**Knot**"-okon halad keresztül. Minden csomópontnál meghatározhatjuk, hogy a görbe milyen szög alatt és milyen ívben érje el. Erre a célra a vezér-





## Spline editor

---

egyenesek szolgálnak. Minden csomóponthoz egy, vagy két vezéregyenes tartozik, attól függően, hogy az a görbe végpontja, vagy közbülső pontja. Lehet olyan csomópont is, amelyen nem halad keresztül görbe, ennek azonban nincs jelentősége.

A vezéregyenesest csak azokban a szerkesztési módokban láthatjuk, amelyekben szabályozni is tudjuk azokat. Ezeket egy szaggatott vonal jelképezi, melynek egyik vége a csomópontban van, a másik pedig egy pontban végződik, amelyen keresztül a vezéregyenesest manipulálni. A görbe mindig úgy éri el a csomópontot lehet, hogy a vezéregyenes legyen az érintője. A vezéregyenes hossza az görbe ívével arányos.

A Spline editor kezelése igen egyszerű, akinek van már egy kis gyakorlata valamilyen vektoros rajzprogrammal, annak teljesen ismerős lesz.

Az editor első két menüje, a Project és Display a szokásos, ezért itt most nem vesztegetünk rá szót.

Az **Object** menüben lévő **Load** és **Save** menüpontok a tárgyak betöltését és kimentését szolgálják. Ezek azonban nem a szokásos poligonális tárgyakat kezelik, hanem a speciális spline formátumúakat. Így kimentett tárgyakat nem lehet más szerkesztőbe betölteni, vagy a színpadon felhasználni. Ha mégis megpróbálnád, hibaüzenet lesz az eredmény. Az elv fordítva is igaz, poligonális tárgyat nem lehet a Spline editorba tölteni.

A **Group**, és az **Ungroup** a tárgyak csoportbakötését és a csoportkapcsolat szétbontását szolgálják a szokásos módon.

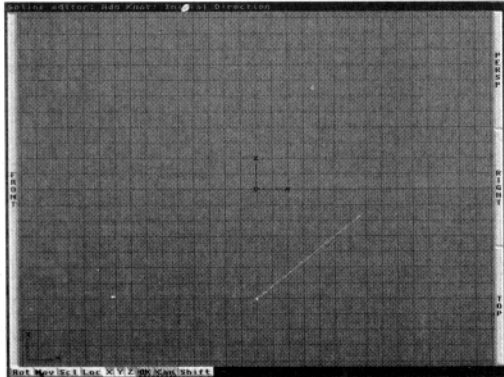
A **Cut**, **Copy** és **Paste** sem szabad, hogy magyarázatra szoruljon, ugyan úgy működik, mint a többi editorban, egész tárgyak clipboardra másolását és onnan történő felhasználását végzik.

Az **Add Axis** segítségével axist adhatunk a szerkesztőhöz. Itt is érvényes, hogy minden tárgy rendelkezik axissal, tehát a tárgyszerkesztést a tengely létrehozásával kell kezdeni.

Amiről eddig szó volt, nem újdonság, most jön a lényeg. Készítsünk egy tárgyat, egy szív alakot, amit térben kinyomunk. Az első lépés tehát a tengely létrehozása. Jelöld is ki rögtön, majd a **Modes** menüből válaszd ki az **Add Knots** módot. Ebben tudunk **"Knot"**-okat, csomópontokat létrehozni. A spline, vagyis a görbe ezeken a csomópontokon halad keresztül. Minden csomópontnál meg kell határozhatjuk, hogy a görbe milyen szög alatt és mekkora ívben érje el.

A görbe létrehozásakor az állapot sorban folyamatosan kiírja a program, hogy éppen milyen műveletet végzünk. Az első üzenet az **"Add Knot: First Point"**. E szerint a görbe első pontját kell éppen meghatározni. Kiklikelj a szív alsó csücskének helyére, innen indítjuk a görbét.

Az állapot sorban megjelenik az **"Initial Direction"** felirat, amely azt jelenti, hogy a görbe kezdőirányát kell meghatározni. Eközben már láthatjuk is az első vezéregyenest. Kiklikelj az alábbi ábrán látható helyre. Most meg kell határozni a következő csomópontot, ezt jelenti a **"Next Point"** üzenet. A pont elhelyezése során folyamatosan láthatjuk a kialakuló görbét, így pontosan beállíthatjuk az alakját, már amennyi ráhatásunk



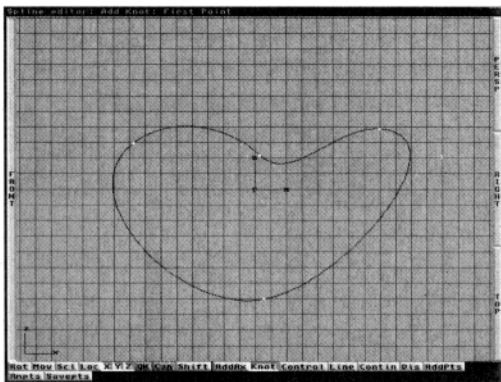
## Spline editor

van arra, ugyan is ez az alak erősen függ az előzőleg meghatározott iránytól.

Ahogy az előbb, most is az irány meghatározása a következő feladat, ez lesz a "**Tangent Direction**". Újból következő pont meghatározása következik..., és ez így megy tovább, míg meg nem unjuk. Két módon lehet befejezni a görbeszerkesztést. Az egyik, hogy az utolsó "Next Point" meghatározásakor a kezdőpontra klikkelünk, ekkor nem is kell irányt meghatározni, hisz az már adva van, ez volt a kezdeti irány. A görbe ezzel bezárult, a szerkesztése automatikusan befejeződik. Természetesen maga a tárgy nem csak egyetlen görbéből állhat, ezért újabb First Point elhelyezésével másik görbét is létrehozhatunk, ez is ugyan ahhoz a tárgyhoz fog tartozni.

A másik lehetőség a görbeszerkesztés befejezésére az **Esc** billentyű lenyomása. Szerkesztő mód váltása csak ezután lehetséges.

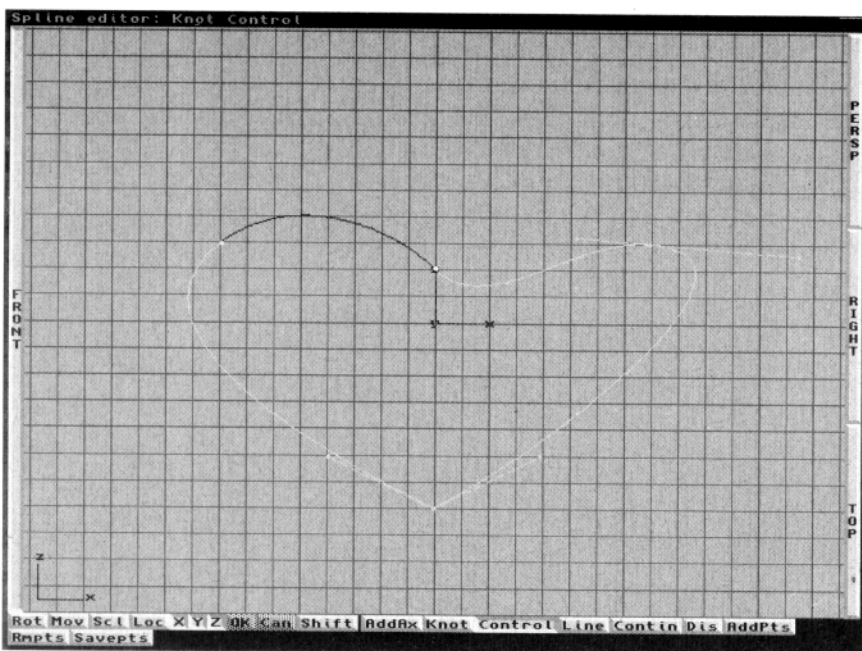
Gyakorlásképpen hozz létre valami hasonló alakot négy csomópontból. Hát, ahogy elnézem, ez nem igazán szív, de ne ess kétségbe, tart még a tanfolyam. Lépj át most "**Knot Control**" módba, amelyben a már elhelyezett csomópontokat és vezéregyeneseket manipulálhatjuk,



ezáltal alakítva a görbét. Ha ráklikkelsz egy csomópontra, megjelennek a hozzá tartozó vezéregyenesek és az egérrel el lehet mozgatni azokat. Maga a csomópont is elmozgatható.

Próbálgasd egy kicsit a lehetőségeket, hogy ráérezz a szerkesztés mikéntjére.

Ha ráuntál a mőkázásra, figyelj ismét rám. Válaszd ki a szív csúcsában lévő knotot és próbáld a vezéregyeneseket úgy elmozgatni, hogy a csúcs kihegyesedjen.

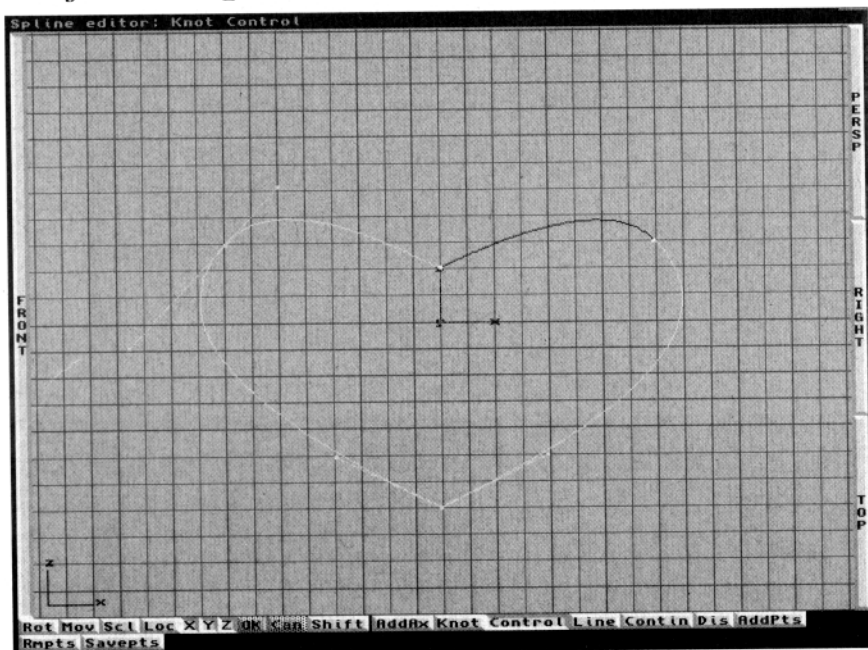


Ha nem megy, ne idegeskedj, alap helyzetben a csomóponton áthaladó görbe folyamatos, a két oldali vezéregyenes egy vonalra esik, csak a hosszuk, azaz a görbének a csomópont két oldalán lévő íve változhat. Válaszd mosd ki a **Functions/Make Discontinuous** pontot. Látszólag nem történik semmi, de ha most elmozdítod a knot valamely vezéregyenesét, a másik egyenes nem mozdul vele. Így már megoldható, hogy a görbe ne legyen folytonos a csomópontban, készítsd el a hegyes

## Spline editor

csúcstól. Ugyan erre a műveletre lesz szükség a szív közepén is. Ennek a műveletnek az ellenkezője a **Make Continuous**, ami ismét folyamatossá teszi a görbét. Van egy harmadik lehetőség is, a **Make Line**, amely úgy rendezi el a knot vezéregyeneseit, hogy a belőle kiinduló görbe egyenes vonalban halad a következő csomópontig. Sajnos az utóbbi műveletnek van egy ici-pici hibája, legalábbis az általam tesztelt változatban, szemben a Make Discontinuous-zel, nem működik Knot Control módban. Használatához át kell lépni Pick Knots-ba és a kívánt csomópont kiválasztása után itt alkalmazni.

Fejezd be a görbe szerkesztését önállóan.



A görbe alak nem tartalmaz egyetlen poligont sem és nincs felülete, létre kell tehát hozni ezeket. Ezt a célt szolgálja az **Objects/Add Points** menüpont.

Kiválasztása után megjelenik az **Extrude/Bevel Data** kérdező, amelyben a tárgy térbeli kinyomásának és az élettöréseinek adatait adhatjuk meg. Nézzük sorba a paramétereiket.

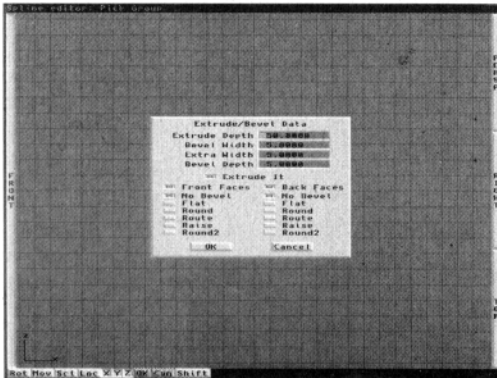
**Extrude Depth** - Az extrudálás mélysége.

**Bevel Width** - Az élettörés szélessége.

**Bevel Depth** - Az élettörés mélysége.

**Extrude It** - Ha ez a kapcsoló be van kapcsolva, csak akkor történik meg a térbeli kinyomás és az élettörés.

**Front/Back Faces** - Ezekkel állíthatjuk, hogy a tárgy kinyomása során az első és a hátsó végére kerüljenek-e záró felületek. Ha nem kapcsoljuk be sem a kinyomást, és egyik felületképzést sem, akkor csak a tárgy körvonalának élei és pontjai jönnek létre, de nem keletkeznek felületek.



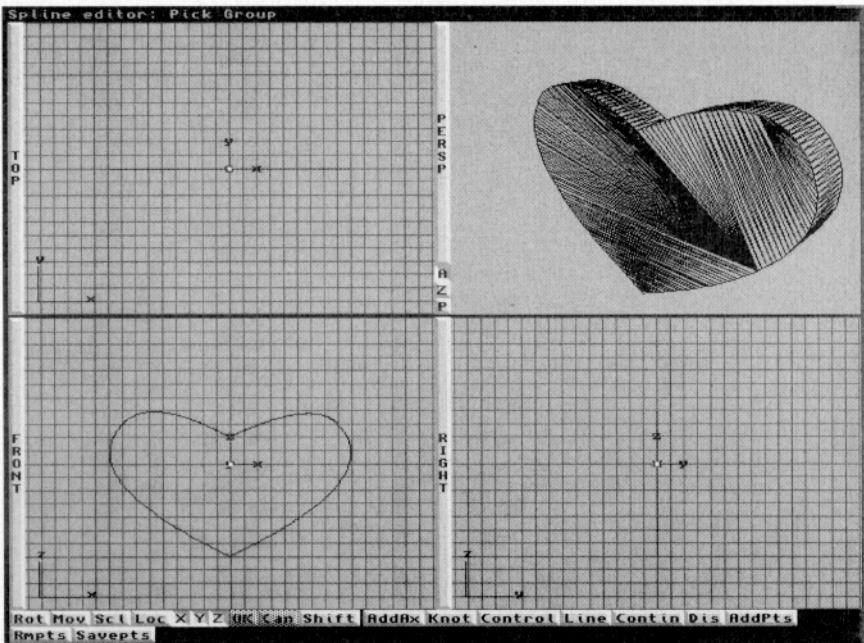
A két oszlopban elhelyezkedő többi kapcsolóval az élettörés módjáról rendelkezhetünk, külön az elő- és a hátlapokra. A **No Bevel** kiixszelése esetén nem lesz az adott oldalon élettörés, a többi kapcsoló használatával különböző algorit-

musok szerint történik meg az éltompítás.

Miután a megfelelő paraméterek beállítása megtörtént, az OK-ra klikkelve a program pontokkal és adott esetben felületekkel látja el a spline tárgyat. Így már az láthatóvá válik a perspektíva nézetben is. A létrehozott poligonális tárgyat a **Save Points** segítségével lehet normál Imagine object formátumban

## Spline editor

kimenteni. A szerkesztőben lévő mindegyik object-nek elkészíthetjük a poligonális változatát és ezeket később is kimenthetjük. Ha egy olyan spline-hoz adunk pontokat az előző módon, amelynek már előzőleg is adtunk, a régi pontok elvesznek és az újak lesznek helyette érvényesek. A poligonális konverzió törlésére használható a **Remove Points** menüpont. Ez nem érinti a spline objectet, csak a hozzá tartozó pontokat.



**Fontos!** Ha egy poligonális konverzióval rendelkező tárgyon további spline műveleteket végzünk, a hozzáadott pontok megmaradnak ugyan, de spline adatok kerülnek a tárgyba, amelyek meghiúsítják a poligonális felhasználást, ezért a konvertálás után azonnal érdemes kimenteni a pontokat.

A tárgyszerkesztéshez rendelkezésünkre áll még négy, idáig nem részletezett funkció a functions menüben, ezek azonban ismerősek lehetnek a Detail editorból.

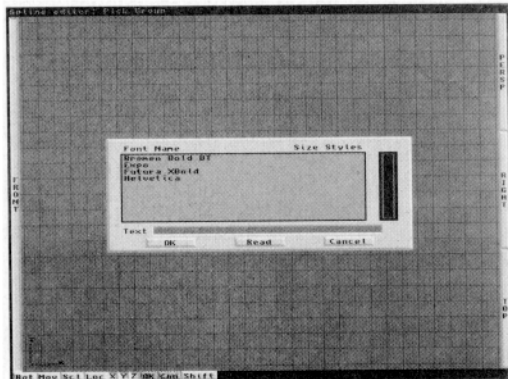
A **Delete** segítségével csoportokat, tárgyakat, vagy knotokat törölhetünk, attól függően, hogy milyen szerkesztő módban vagyunk.

A **Join** funkció spline tárgyak egymásba fűzését szolgálja, pontok összekapcsolását nem tudjuk ezzel elvégezni. Egy pontot csak akkor lehet összekapcsolni egy másikkal, ha az végpont, vagy görbe nélküli, önálló pont. Egy csomóponton csak egy görbe mehet keresztül. Az összekapcsolás nagyon egyszerű, **Add Knot** módban először az egyik, majd a másik pontot kell kiválasztani az új görbe csomópontjainak. Irányt meghatározni nem mindig kell, hisz az a két pont által általában már adott. Ha ez esetleg nem megfelelő, később lehet módosítani rajta.

A **Fracture**-ral a **Pick Knots** módban kiválasztott görbéken helyezhetünk el újabb csomópontot. Ez a művelet nem változtatja meg a tárgy alakját.

A **Break** megszünteti a kiválasztott knotból induló görbét, megszakítva ezzel az alak folytonosságát.

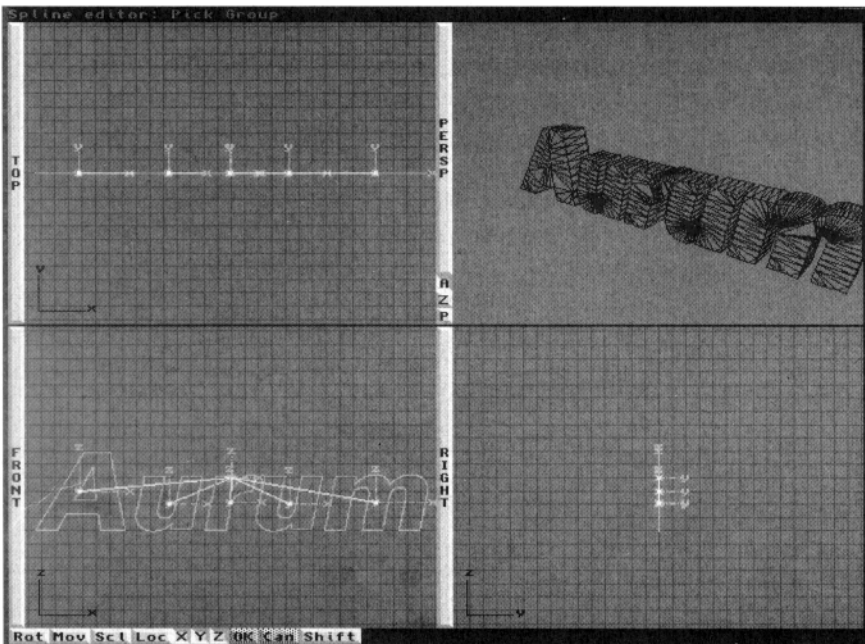
A fejezet elején írtam, hogy a Spline editorban font objecteket is létrehozhatunk. Ennek mikéntje igen egyszerű, csak ki kell választani az **Objects/Load Font** menüpontot. Megjelenik egy kérdező, amelybe a Read kapcsolóval tudunk **Adobe Type 1**-es fontot beolvastatni.





## Spline editor

A fájlselektorban nem csak a font direktorit kell megadni, hanem egy fontfájlt is. Az OK után a program felkutatja az adott könyvtárban lévő fontokat és ezek nevét megjeleníti a kérdező listájában. A listából választhatunk ki egy fontot, a megjelenítendő szöveget pedig a **Text** input mezőbe írhatjuk. Az OK hatására a program a megadott fontkészletből előállítja az objectet, amely spline körvonalként jelenik meg a szerkesztőben. Ezt azután a korábban említett módon alakíthatjuk térbeli objectté és láthatjuk el pontokkal, készíthetünk belőle poligonális tárgyat.



Nos, ennyi a Spline editor rövid története, egyszerű, de nagyszerű.

## Action editor

Az Action editor mindössze néhány újdonságot tartalmaz a 2.0-s változathoz képest, azonban ezek mindegyike igen értékes és jól használható. A legfontosabb újdonság a morfózisok terén található, ez a Detailban már emlegetett state animációval van kapcsolatban. Ezentúl a morfózis elkészítéséhez nem kell több tárgyat létrehozni, hanem az egyes részállapotokat a tárgyak önmaguk tartalmazzák, ami kényelmesebbé és gyorsabbá teszi a munkát.

Nem túl jelentős, de azért hasznos bővülés, hogy egyszerre akár négy efektet is rendelhetünk a tárgyakhoz. Megjelent egy új effekt típus is, a **Global Effect**, amely tagjait nem a tárgyakhoz, hanem a Globals-hoz kell rendelni, Hatásuk az egész képre kiterjed, általában a kész képet módosítják és nem a számítás menetét befolyásolják. Részletesen ezekről egy külön fejezetben olvashatsz.

A tárgyak beállítását könnyíti, hogy különböző rétegekre tehetjük azokat. Ezek nem térbeli rétegek, hanem teljesen logikaiak. A Stage editorban meghatározhatjuk, hogy mely rétegeken lévő tárgyak jelenjenek meg a szerkesztőben, így sok tárgy színre kerülése esetén is átlátható marad a szerkesztő és a képfrissítések is gyorsabbak. A renderelés során minden réteg tárgyait figyelembe veszi a program, arra tehát ez nincs hatással. Azt, hogy egy tárgy mely rétegre kerüljön, a betöltése során határozhatjuk meg, de ez később könnyedén megváltoztatható.

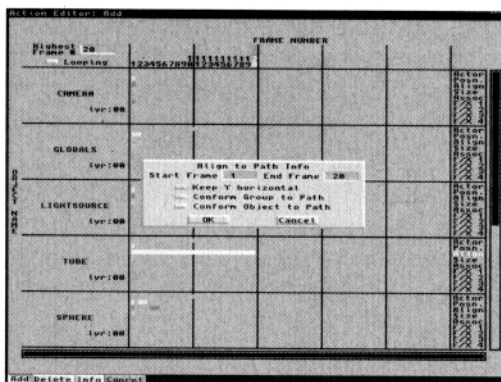
Az ismétlődő paraméterek beállítását segítik a **Copy, Cut, Paste** funkciók. Ezekkel beállított szakaszok paramétereit másolhatjuk vágólapra, majd onnan átvihetjük más pozícióba, vagy más tárgyra.

## Action editor

Az animációk loopolását segíti egy kapcsoló, a **Looping**. Ezt aktiválva a program úgy számol, hogy az animáció utolsó képkockája azonos az elsővel, az egyes mozgásokat és átalakulásokat is ennek megfelelően számolja ki. Használata esetén ügyeljünk arra, hogy a tárgyak az animáció végére eredeti helyzetükbe térjenek vissza, ha nem ez teljesül, a program figyelmeztet.

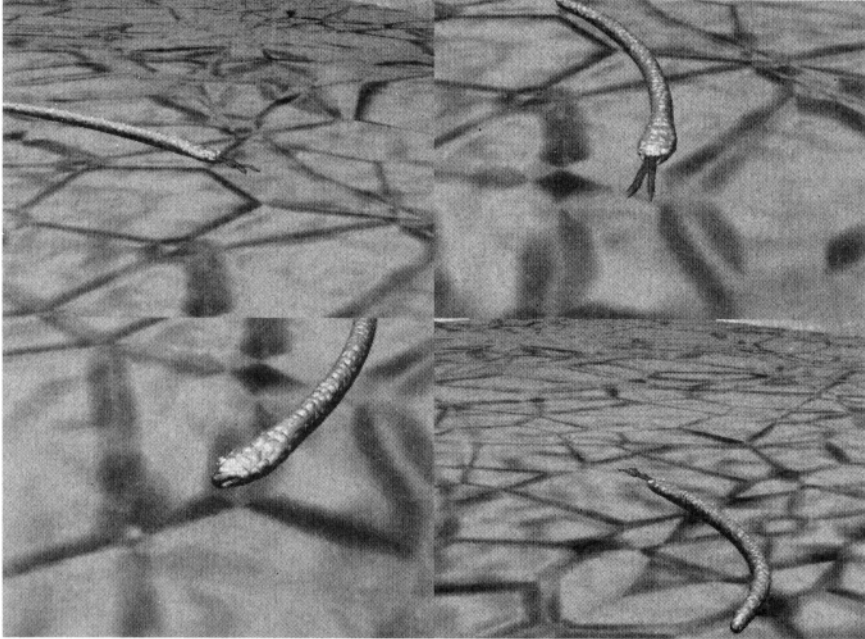
Aprócska változás, hogy a Hinge funkciót most **Associate**-nek hívják, lényeges különbség azonban nincs közöttük.

Az utat követő tárgy **Alignment** beállításánál is megjelent két új lehetőség, két kapcsoló, amely összefügg egymással. Hozz létre egy görbe utat és egy több szeletből álló Tube tárgyat. Állítsd be úgy a tengelyét,



hogy az úttal egy irányba nézzen. Tegyéél az Alignment sorba is felsorolójelet arra a szakaszra, amelyen a tárgy követi az utat, válaszd az **Align to Path** módot. A **Keep Y** horizontal jelentése a szokásos, a tárgy megtartja Y körüli orientációját, és nem az út ilyen paraméterét veszi át. Az újdonság a következő két kapcsoló. Ha egyik sincs kipipálva, akkor a hatás a szokásos, a tárgy X és Z tengelye mindig az út megfelelő tengelyeivel azonos módon fog beállni. Ha a **Conform Group to Path** opciót használjuk, a csoport, miközben követi az utat, nem csak az orientációját változtatja annak megfelelően, hanem minden tagja is megpróbálja azt követni,

mint egy hullámvasúton haladó szerelvény kocsijai. A **Conform Object to Path** hatására a csoport összülője nem csak az orientációjával, hanem alakjával is követi az utat. A kialakuló hatást szemléltetik az alábbi ábrák:



Utóbbi funkcióval a kigyózó kígyó létrehozása már nem is olyan nagy dolog. Ráadásul az út alakját is folyamatosan változtathatjuk statek segítségével, akár egy normál tárgyét, ezzel a kígyó összehúzódását-megnyúlását is utánozhatjuk. Minél több szegmensből áll a tárgy, annál jobban tud hozzáidomulni az úthoz.

### Tárgy betöltése

Lássuk hogyan tölthetünk be a megváltozott körülmények között egy tárgyat az Action editorban. Válaszd ki a menüből, vagy az User Gadgettel az **Add** módot, majd a legutolsó tárgy sorában klikkelj egyet-egyet azon kockák alatt lévő **Actor** sorba, amely animációs szakaszra a tárgyat be szeretnéd tölteni. Megjelenik a szokásos fájl szelektor, itt válaszd ki a betöltendő tárgyat. Az OK, vagy az Enter lenyomása után jön az első különbség, megjelenik egy input mező, amiben annak a rétegnek a számát kell megadni, amelyikre a tárgyat tölteni szeretnénk. A megfelelő réteg számának beírása és az Enter lenyomása után megjelenik az **Object File Info** kérdező. Ebben még módosíthatjuk a tárgy színre kerülésének szakaszát a **Start** és **End Frame** input mezőkben, vagy a **Filename** sorában megváltoztathatjuk a fájl nevét. Utóbbi műveletet egy fájl szelektor segíti, amelyet a **Browse** gombra kattintva hívhatunk elő. A fájl neve alatt egy kisebb input mezőt találunk, ebben lehet kiválasztani, hogy a tárgy mely állapotában legyen jelen a kért szakaszon. Ide a megfelelő state nevének kell kerülni. Ennek beírását is segíti egy szelektor, amely a tárgyban definiált statek listáját jeleníti meg. Ha a tárgy tartalmaz stateket, akkor sem kötelező használni, ekkor a tárgy abban az állapotában vesz részt az animációban, amelyben kimentéskor volt, még akkor is, ha ez nincs definiálva egyetlen state-ben sem. Ha ugyan ez a tárgy, ugyan ebben az Actor sorban ezt a szakaszt közvetlenül megelőzően használva van, a tárgy ez alatt a szakasz alatt az előző állapotából folyamatosan, a szakasz végére alakul át az itt megadott állapotába. Ez a morfózis létrehozásának módja az új verzióban. Ha azonos

Actor sorban több különböző tárgyat töltünk be, akkor azok meg kell hogy feleljenek az Imagine normál morfózis szabályának, tehát azonos számú pontból, élből és felületelemből kell állniuk.

Folytassuk tovább az Object File Info kérdezővel való ismerkedésünket. A **Number of cycles to perform** nem változott, a cycle object által az adott szakaszon elvégzendő ciklusok számát jelenti, amely lehet tört szám is.

Az **Initial Cycle Phase** szintén a régi, a cycle object kezdő fázisát jelenti az egész ciklus hosszához képest.

A **Reverse cycle motion** hatására a cycle object visszafelé végzi a ciklus lejátszását.

A **Spline Interpolation** egy új kapcsoló, használata esetén az átalakulást végző tárgyak két kulcspont között spline, azaz görbe vonalú mozgásokat végeznek és nem szögleteseket. Ez most nem a pozícióváltozásokra vonatkozik, hanem az átalakulás sebességfolyására és egyéb dolgokra, simább morfózist létrehozva.

A **Discontinuous Knot** szintén új opció, az előző kapcsoló aktiválása után van hatása, lehetővé teszi, hogy az átalakulási görbe a csomópontokban éles irányváltást tegyen. Például egy V alakú mozgást leíró stage sorozat esetén a Spline opciót bekapcsolva a tárgy a csomóponthoz lágyan közelít, görbe íven halad át azon. Ha a Discontinuous opciót is használjuk, akkor a V csúcspontján a mozgás élesen vált irányt.

**Vel. Scaling #0/#1** - Ez egy új lehetőség, a morfózis lefolyásának sebességét tudjuk vele szabályozni. A **#0** az átalakulás sebessége kezdetben, a **#1** pedig ugyan ez, de az átalakulás végén. Ha mindkét érték 1.0, az átalakulás egyenletes sebességgel zajlik le. Ha a **#0** a kisebb, akkor az átalakulás lassabban

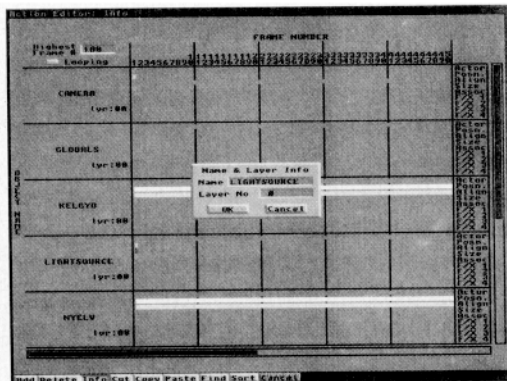
## Action editor

indul meg, a végére gyorsít, ha a #0 a nagyobb, akkor fordítva. Ha mindkét érték kisebb, mint 1, az átalakulás lassabban indul, középre érve gyorsul, majd a végére ismét lelassul. Ezzel elérhetjük, hogy a morfózis lágyan menjen végbe, ne pedig hirtelen indulva és befejezve.

Utóbbi négy opciót megtalálhatjuk a **Position**-nál, az **Alignment**-nél és a **Size**-nál is. Hatásuk ezekkel azonos, kivéve, hogy nem az átalakulás, hanem a pozícióváltozás, illetve az irány- és méretváltozás lefolyását szabályozzák.

Az előbb ismertetett kérdezők akkor is megjelennek, ha a tárgyat a **Stage** editorban az **Objects/Load** segítségével töltjük be. Mivel ezek teljesen egyformák, ezért leírásukat ott már nem ismétlem meg.

Említettem, hogy a tárgy könnyedén áthelyezhető egyik rétegről a másikra. Ennek módja, hogy **Info** módban a tárgy nevére kattintunk (nem az Actor nevére, hanem az első oszlopban szereplő névre). Megjelenik egy kérdező, melyben két input mező található. Az egyikben az a név olvasható, melyet az Attributes kérdezőben adtunk a tárgynak, ezen a néven nevezzük a tárgyat az Action, Stage és Detail editorokban. A nevet átírva az Action és Stage editorokban ezen az új néven lesz nyilvántartva az a tárgy. Külön Rename funkciót nem is találunk.



A másik paraméter a **Layer No**, ami a tárgy layerének száma, ezt megváltoztatva a tárgy átkerül az új rétegre. A használt réteg száma egyébként állandóan látható az Action editorban a tárgy neve alatt jobbra.

### Beállítások másolása

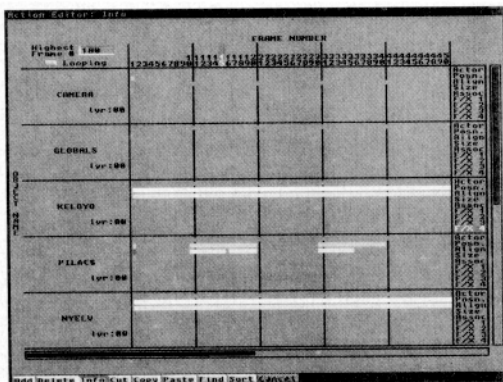
Másik kényelmi funkció a rétegek mellett, a clipboard használatának lehetősége. Több tárgy azonos beállítása esetén nagy segítség, hogy bizonyos felsorolójel szakaszokat bemásolhatunk a clipboard-ra, majd onnan bárhova visszamásolhatjuk (persze csak ha van értelme, például pozíció az alignment-hez nem rakható). Nézzünk erre egy példát, több tárgyat azonos pozícióba kell hozni. Válaszd ki a **Copy** módot, majd annak a tárgynak a pozíciójelen egy-egy klikkel jelöld ki a kiemelő szakasz kezdetét és végét, amelynek a helyzetét át akarod másolni. A kijelölt szakasz bemásolódik egy átmeneti tárolóba, az üzemmód pedig átvált **Paste**-ra, amelyben a vágólapon lévő paramétereket helyezhetjük le. Ha most egy másik tárgy Position sorában (de lehet ugyan ez a tárgy másik animációs szakaszán is) klikkelsz egyet, az előbb bemásolt szakasz oda kiíródik. Ennek azonban feltétele, hogy a kirakandó szakasz beférjen az új helyre, és típusuk megegyezzen. Nem csak egy paraméter másolható át ilyen módon, hanem egyszerre több is. Nézzünk erre is egy példát. Nem csak a pozíciót, hanem az alignmentet is át kell másolni, ekkor az első klikk a pozíció sorában az átmásolandó szakasz kezdetén kell, a második pedig az alignment sorában a szakasz végén. Ha ezután a célnál Paste módban klikkelünk egyet, nem csak a pozíció, hanem az irány felsorolójel is odakerül.



## Action editor

Amikor a vágólapra másolunk, a felsorolójeleknek nem kell folyamatosnak lenni, több, akár össze nem függő részből is állhatnak. Lényeges azonban, hogy csak azok a felsorolójel-szakaszok kerülnek a vágólapra, amelyeknek a kezdete beleesik a kijelölt szakaszba, valamint fontos az is, hogy a felsorolójelek egyben másolódnak, még akkor is, ha csak egy kis szakaszt jelöltük ki.

Tegyük fel, a 10-20 képkockák között van egy pozíció, valamint a 10-15 és 16-20 kockák között két irány felsorolójel. Copy módban az 5-12 szakaszokat jelöljük ki és másoljuk vágólapra, úgy, hogy a pozíció és az irány jelek is felkerüljenek. Az eredmény az lesz, hogy a vágólapra kerül a teljes 10-20 pozíciójel szakasz, valamint a 10-15 alignment jel. A 16-20 alignment nem kerül a vágólapra, mert a pozíció kibővülése 20-ig, nem vonatkozik másra, csak a megkezdett szakaszra.



Ha ezután pl. a 30. képkocka alá klikkelünk, (mindegy melyik sorba, hisz a pozíció akkor is a pozíció alá fog kerülni) a pozíció felsorolójel a 35. képkockától kezdve kerül fel. Ennek az az oka, hogy a felmásolásakor az 5-9 szakaszok is vágólapra kerültek, még ha itt nincs is felsorolójel.

A vágólapra másolás másik módja a **Cut** funkció, ez annyiban különbözik a Copy-tól, hogy a felmásolt szakaszokat törli eredeti helyükről. A törlés, a másoláshoz hasonlóan a teljes megkezdett szakaszt érinti.

A fent részletezett vágólapos funkciók igen hasznosak például ismétlődő mozgások definiálásánál. Tegyük fel, hogy egy hosszú animáció során egy inga lengését kell beállítani. A lengést az inga két végpozíciójának state-tel történő meghatározásával állíthatjuk be. Elég azonban egyetlen lengési ciklust beállítani az Action editorba, ezt copyval vágólapra tehetjük, és ugyan azon actor sorban tetszőleges számban visszamásolhatjuk, de akár új tárgyként is felvihetjük.

### Morfózisok készítése

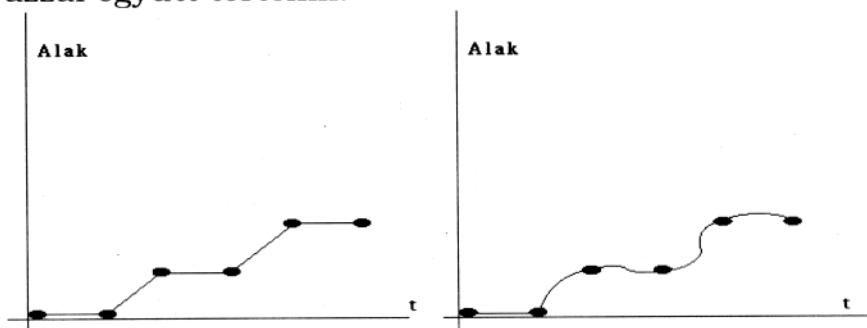
A korábbi programváltozatban átalakulást úgy készíthettünk, hogy azonos Actor sorba két különböző tárgyat töltöttünk, közvetlenül egymás után. Ezek között a program a második tárgy szereplési szakaszának végére az első tárgyat átalakította a másodikra. Lényeges megkötés, hogy a két tárgynak azonos számú pontból, élből és felületből kell állnia.

Ez a lehetőség még megvan, de a program felkínál egy jobbat is, a már sokszor emlegetett statek használatával. Ebben az esetben is egymás után, azonos Actor sorba kell kétszer betölteni a tárgyat, de két különböző state-t kell választani. Az átalakulás e két state közt megy végbe. Mivel a kiindulási és a cél object azonos, nem jelent problémát az előbbi megkötés, hisz automatikusan teljesül.

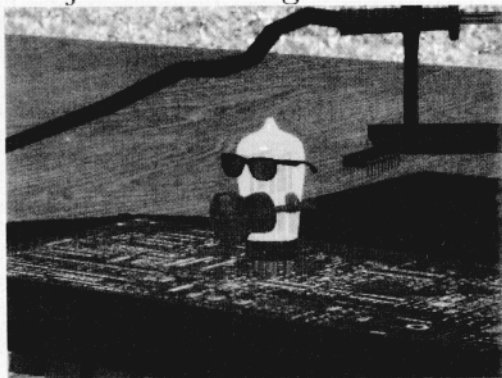
Hosszabb, több átalakulást tartalmazó beállításnál a kívánt eredmény korrekt eléréséhez be kell tartani néhány szabályt. Tegyük fel, van egy tárgyunk, három state-tel. Ezek neve S1, S2, S3. Ezek között szeretnénk olyan átalakulást, hogy az első 10 kockában az S1 állapotban van a tárgy, a 11-20 képek között átalakul S2-be, amely állapotot 21-30 képek között fenntart, majd 31-40 között tovább alakul S3-

## Action editor

ba, amelyet fenntart 41-50 képkockák között. Ha az átalakulásokra nem használjuk a Spline Interpolation-t, akkor elegendő öt felsorolójel az Actor sorban, azt kapjuk, amit vártunk. Ha azonban lágy átalakulásokat szeretnénk, már bonyolultabb a helyzet, a fenti megoldás esetén a tárgy akkor is végez alakváltozást, amikor nem kellene. Mi ennek az oka? Nézd meg a következő két grafikon. Az első azt az esetet mutatja, amikor az átalakulás Spline Interpolation nélkül, a második pedig azt, amikor azzal együtt történik.



Ennek kiküszöbölését szolgálja Discontinuous Knot kapcsoló, amely megszünteti a görbe folytonosságát a csomópontokban, így az oda befutó görbe alakja nem befolyásolja a kiinduló görbét.



### 3D hatás és mélységélesség

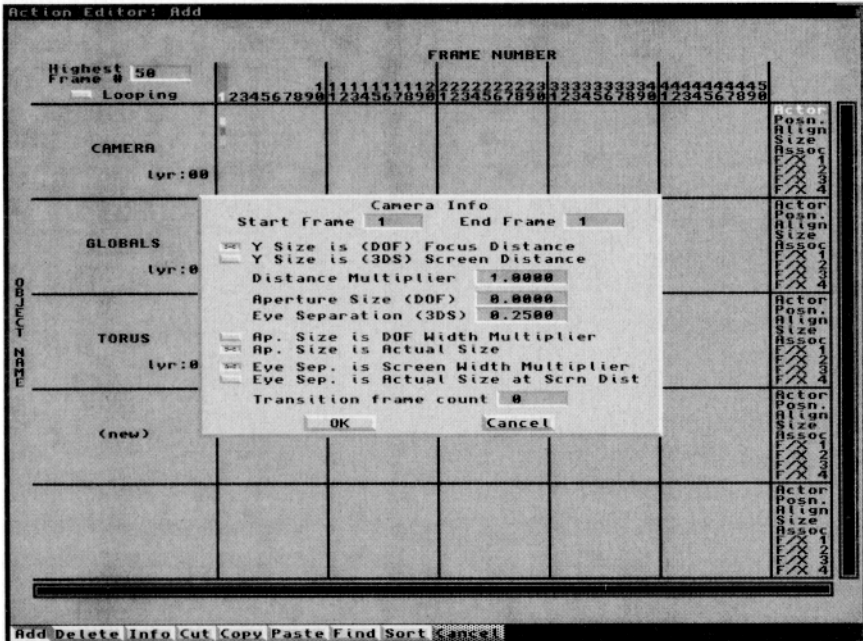
Az Imagine lehetőséget ad 3D sztereó képek készítésére, amelyeket egy speciáli szemüveggel megtekintve, valódi három dimenziós hatást érhetünk el. Ezt már a korábbi program változat is modellezni tudta, hogy mégis szó esik róla, annak az az oka, hogy a beállításának módja megváltozott.

A 3D képek generálását a **Project** editorban a **Stereo 3D** opció aktiválásával kapcsolhatjuk be, de a paramétereit az **Action** és **Stage** editorban kell beállítani. Azért, hogy az információkat ne három fejezetből kelljen összeszedni, itt egyben olvasható minden, amit a 3D beállításával kapcsolatban tudni kell.

Mielőtt belemélyednénk a dolgokba, nézzük hogyan jön létre a három dimenziós hatás. A program minden képet két részből épít fel. A két kép két különböző kameraállásból készül el, a kamerák vízszintesen kicsit el vannak tolva egymástól mint ahogy az ember szeme. Az elkészült két félképet soronként felváltva egymásba fűzi a program. Ezután egy speciáli szemüvegre van szükség, amelynek üvegei LCD kristálybevonatúak. Egy elektronika felváltva, a video félképek ütemében hol az egyik, hol a másik üveget sötétíti el, minek hatására mindkét szem csak a neki megfelelő képet látja, így kialakul a valódi 3D látvány.

A hatás modellezéséhez két dolgot kell ismernie a programnak, a szemek távolságát a megfigyelt tárgyak síkjától, vagyis azt, hogy milyen messziről tekintünk, valamint a szemtávolságot. Előbbi értéket a kamera Y tengelyének hosszával, utóbbit a **Camera Actor** Infon-n keresztül tudjuk szabályozni.

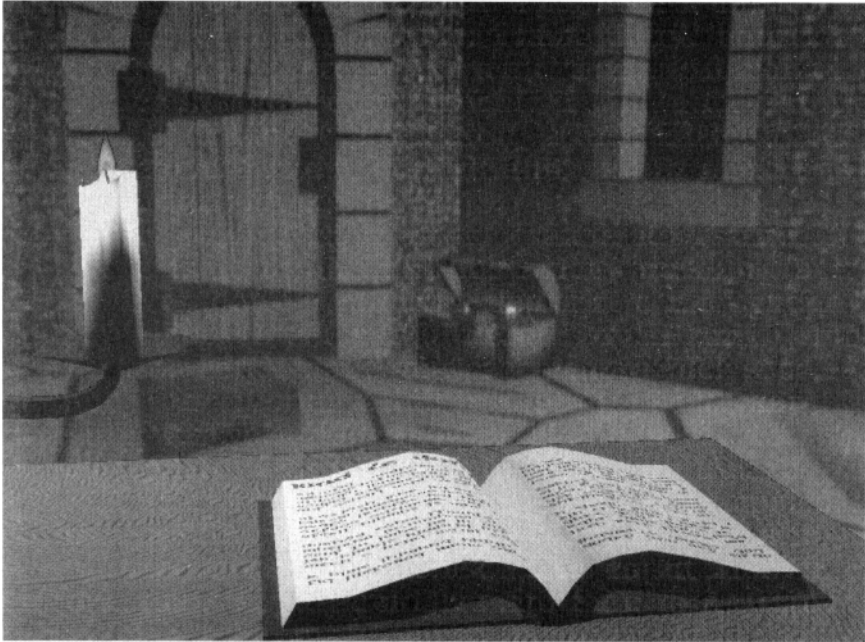
# Action editor



Ahhoz, hogy a kamera Y valóban a 3D távolságot szabályozza, a Camera Info-ban az **Y Size (3DS) Screen Distance** kapcsolót aktiválni kell.

Az Y hossza az a távolság, amelyre a szem tekint. A szemtávolság értéke az **Eye Separation** mező értéke. Ez nem abszolút érték, hanem származtatott. Alapja lehet az előbbi távolság, vagy a kép szélessége. Hogy melyik, azt az **Eye Sep. is...** kapcsolók döntenek el. A **Screen Width Multiplier** aktiválása esetén a szemtávolság a képszélesség, vagyis a nézőtávolságban lévő sík képernyőn megjelenő méretének, az említett input mezőben megadott szorzata lesz. A **Size at Scrn Dist** kapcsolót aktiválva, a szemtávolság a nézőtávolságból származik, az Y tengely hosszát változtatva, az is változik.

A másik hatás, amelyet a Camera Info-ban szabályozhatunk, a mélységelesség. Az Imagine képes utánozni a fényképezőgép mélységelességét, vagyis hogy a képen csak egy bizonyos távolságban lévő tárgyak látszódnak élesen ettől a távolságtól közeledve, vagy távolodva egyre életlenebbnek látszanak a tárgyak. Az élesség csökkenése a kamerához közeledve erőteljesebb, mint attól távolodva.



Az a távolság, amelyre a kamerától lévő tárgyak tökéletesen élesnek látszanak, a kamera Y tengelyének hosszával szabályozható. Hohó, hisz az a 3D hatásnál már foglalt, kiáltanak fel sokan. Igazuk van, a kamera Y tengelyhosszát kétszeresen is felhasználja a program. Hogy ez éppen mit határoz meg, a kérdező tetején lévő két kapcsoló dönti el. Ha a mélységelességre szeretnénk használni az Y

## Action editor

---

hosszt, akkor az **Y Size is (DOF) Focus Distance** kapcsolót kell aktiválni. Igen ám, de ekkor hogy határozzuk meg a nézősíkot a 3D hatás számára, vagy nem lehet e két dolgot együtt használni?, hangzik el a jogos kérdés. De lehet használni, megadhatjuk, hogy a másik funkció az Y hosszának mekkora százalékát használja. Ez az arány a **Distance Multiplier** input mező értéke.

Tegyük fel például, hogy a kamera Y hossza a 3D távolságot határozza meg közvetlenül, hossza 100 egység. Azt szeretnénk, hogy a fókusz távolság 200 egység távolba essen. Ekkor a Distance Multiplier értékének 2.0-t kell adni. Látható tehát, ha Y hosszát változtatjuk, mint a 3D, mind a fókusz távolság változni fog.

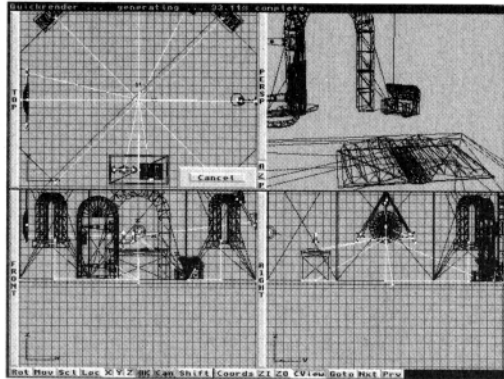
Nem elég azonban csak a fókusz távolságot tudni, azt is ismernünk kell, hogy mekkora az a távolság, amelyen belül még éles a kép. Ezt egy fényképezőgépen a rekeszszel tudjuk szabályozni, minél jobban zárjuk, annál nagyobb lesz a mélységélesség, az az annál nagyobb távolsághatárok közt kapunk a tárgyról éles képet. A Imagine-ben ezt a **Aperture Size (DOF)** input mező értékével tudjuk szabályozni. Kétféle módon származtathatjuk a mélységélességet, az **Ap. Size is DOF Width Multiplier** aktiválása után a kép szélességének és az input mező értékének szorzata adja, míg az **Ap. Size is Actual Size**-t aktiválva az input mező abszolút értéke Imagine egységben.

Ne feledkezzünk el arról, hogy a kamera látószögének meghatározásában is szerepe van az Y tengelyének, ha valamely fenti hatás beállítása miatt változtatni kell a kamera Y tengelyen, de nem akarjuk, hogy a látószög megváltozon, azonos mértékben kell változtatni az X méretét is!

## Stage editor

A Stage editor szorosan összefügg az Actionnal, ezért a leírása során sűrűn fogok hivatkozni arra. A Stage lényege, szerkezete, hasonlóan a program többi részéhez, nem változott, azonban itt is megjelent

jó néhány igen hasznos újítás. A szerkesztést segítő, nem csak a kamera nézőpontjából nézhetjük meg a színpadot a perspektíva nézetben, hanem egy tetszőleges irányból, mint a Detailban, vagy akár valamely más tárgy,



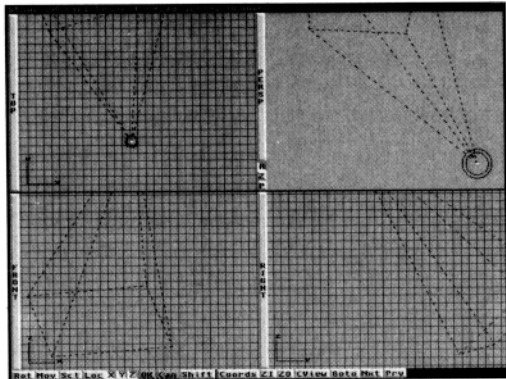
vagy fényforrás szemszögéből is. Utóbbi funkcióval könnyedén beállíthatjuk az irányított fényű lámpáinkat, hogy ténylegesen oda világítsanak, ahová szeretnénk. Ha valamely tárgy nézőpontjából kérjük a perspektíva nézetet, olyan képet kapunk, mintha a kiválasztott tárgy tengelypontjából +Y irányában néznénk, úgy, hogy a +X jobbra esik. Ebből adódóan +Z van felfelé, akár a kameránál, de ez vonatkozik a fényforrások nézetére is.

A kamera és a fényforrások beállításával kapcsolatos másik újítás, hogy megjeleníthetjük irányukat és látószögüket. (Fényforrás esetén csak ha irányított fényű). Ez azt jelenti, hogy a megfelelő irányokat és látószögeket szaggatott vonalak jelölik mindhárom síknézetben, de az új perspektíva mód esetén a negyedik nézetben is.



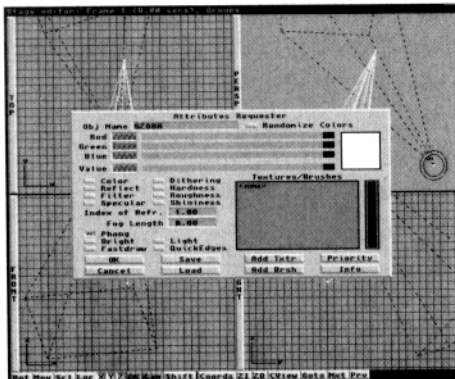
## Stage editor

Harmadik újítás a perspektíva nézettel, hogy nem csak a síknézeti ablakokban állíthatjuk be a tárgyakat, hanem egy új megjelenítési módba kapcsolva a perspektíva nézetben is, ami jelentősen megkönnyíti a térbeli munkát. Ha ezt bekapcsoljuk, akkor a kamera és a fényforrás látószögét jelképező szaggatott vonalak is látszanak ebben a nézetben.



A szerkesztőablakok hátterébe a Detail editorhoz hasonlóan háttérképeket tehetünk, amelyek segítik a beállítást. Minden nézethez, még a perspektívához is külön kép tartozhat. Nem csak önálló képet, hanem **IFF-Animbrush**, vagy **FLI** animáció egy kockáját is betölthetjük a háttérbe.

A Quickrender gyorsítását szolgálja, hogy kijelölhetünk a perspektíva képen egy négyzet alakú területet, és csak az azon belüli részt számolja a



program, nem az egész képet. Ez főleg olyankor jó, ha csak egy részlet beállítására vagyunk kíváncsiak.

Hasznos dolog, hogy a tárgyak attribútumait immár a **Stage** editorban is megváltoztathatjuk, innen is hívni lehet az **Attributes** kérdezőt, az

**Objects/Attributes** menüponttal. Ez a kérdező teljesen azonos a Detailban megjelenővel. Amikor a tárgy attribútumainak beállításával elkészültünk, a program azonnal felajánlja a tárgy kimentésének lehetőségét. Ha ezt elmulasztjuk, a Stage editorból kilépve, vagy másik képkockára átváltva, amikor is a tárgyak újra olvasódnak a lemezről, a beállítások elvesznek!

Amikor ugyan azt a tárgyat több példányban, de azonos beállításokkal kell felhasználni, fárasztó, hogy egyenként mindegyiket be kell állítani. Ezen a nehézségen segít a **Objects/Clone**, amely klónozza a kiválasztott tárgyakat, azok minden tulajdonságával és beállításával együtt. A klónok kissé más pozícióban jelennek meg, mint az eredeti tárgyak, innen azonban bárhova elmozgathatók.

A képkockák közötti navigálást segíti a **Frame** menü két új menüpontja, a **Next Brk** és a **Prev Brk**. Ezeket használva a kiválasztott tárgy következő, vagy előző töréspontjára, vagyis valamely felsorolójelenek kezdetére/végére ugorhatunk. A menüpont használata után megjelenik egy kérdező, melyben kiválaszthatjuk, hogy melyik paramétertípus következő/előző töréspontjára szeretnénk ugrni.

Ha a kiválasztott paramétertípusnak nincs abban az irányban több töréspontja, akkor nem történik semmi, ha van, akkor a megfelelő képkockára ugrik a Stage editor.

Az általános áttekintés után nézzük meg az egyes újdonságokat részletesen is.

Az említett új perspektivikus módra átkapcsolni, melyben a síknézetekhez hasonlóan tudjuk a tárgyakat beállítani, a **Display** menü **NewMode** pontjának kipipálásával tudunk. Ekkor a perspektíva nézetben is megjelennek a tárgyak tengelypontjai, amelyek ott eddig nem voltak láthatóak, valamint láthatóvá

válnak a fényforrások és a kamera. Ezután a perspektíva nézetet is úgy használhatjuk, mint a síknézeteket, a tárgyakat kiválaszthatjuk, mozgathatjuk, forgathatjuk, stb. Normál módra visszatérni a **Wireframe**, **Solid**, vagy **Shaded** valamelyikének kiválasztásával lehet. Ez az új megjelenítési mód működik nem csak független, hanem kamera, fényforrás, vagy tárgynézetben is.

Ha már a fényforrás és tárgynézetnél tartunk, ez szintén egy újdonság. Lehetőség van arra, hogy a szerkesztőablakban ne csak egy független nézetben, vagy a kamera szemszögéből tekintsünk a színpadra, hanem valamely fényforrás, vagy tárgy szemszögéből. Ezzel nagyon leegyszerűsíthető az irányított fényű fényforrások beállítása, hiszen azok épp oda fognak világítani, ahová a nézetükkel látnak. Természetesen nem csak irányított fényű fényforrásoknál van lehetőség erre a nézetre, de egyéb esetben a lámpa minden irányban sugároz fényt, nem csak oda, ahová néz.

Hogyan válthatunk át erre a perspektíva nézetre? A **Display** menü **Lightsource View**, vagy **View from Object** pontjainak használatával. Mindkét esetben megjelenik egy kérdező, benne a színen lévő tárgyak listája, amelyekből kiválaszthatjuk a megfelelőt. A **Lightsource** kérdezőjében csak a "**Stage**" fényforrásokat, vagy a **Track Axis**okat választhatjuk ki, normál tárgyakat még akkor sem, ha fényforrás attribútummal rendelkeznek. A kiválasztás után a perspektíva nézet úgy alakul, mintha a kiválasztott tárgy tengelypontjából a tárgy +Y irányba tekintenénk, miközben +X jobbra esik, következésképp +Z van függőlegesen felfelé. Ha a **NewMode** van aktiválva, akkor a tárgyakat az új nézetben térben is manipulálhatjuk.

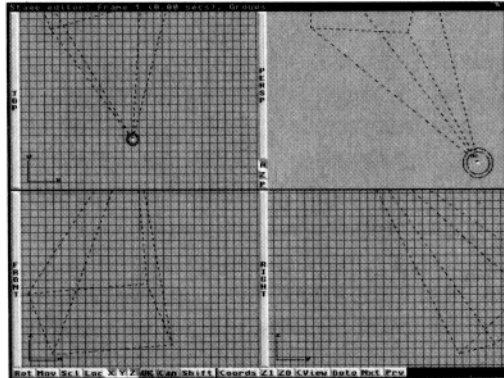
Lényeges különbség a Lightsource View és View from Object között, hogy előbbi esetben a nézőpont ténylegesen kapcsolódik a fényforráshoz, elmozgatva azt, a nézőpont is mozog, míg utóbbi esetben a független perspektíva nézet áll be úgy, mintha a tárgy tengelypontjából néznénk a színpadra, de a tárgyat mozgatva a nézőpont nem változik, az marad az előző helyén. Ha a Lightsource View kérdezőjében egy Track Axis-t választottunk ki, az is így viselkedik.

A tárgy és a fényforrás látószögét az Y/X tengelyviszonya határozza meg, hasonlóan az irányított fényű fényforrás fényének nyílásszögéhez, vagy a kamera látószögéhez.

A kamera és a lámpák beállítását az is segíti, hogy megjeleníthetjük a nézőirányukat és látószögüket. A **Display** menü **Camera Lines** és **Light Lines** pontjait kipipálva a ka-

mera és a fényforrások irányát és látószögét szaggatott vonalak fogják mutatnia a síknézetekben és bekapcsolt NewMode esetén a perspektívanézetben. Fényforrás esetén csak akkor rajzolódik fel ez a szaggatott vonal, ha az irányított fényű, hiszen ellenkező esetben a fény minden irányban egyformán terjed.

A Detail editorban már láttuk, hogy a perspektívanézet függetleníthető a síknézetektől, a numerikus beállításra ugyan úgy van lehetőség. Ezek a beállítások csak akkor éreztetik a hatásukat, ha nem kamera, vagy fényforrás nézetben vagyunk, hisz



## Stage editor

---

akkor a kamera, vagy a fényforrás tengelye adja a nézőpontot, nézetirányt és a látószöget. Ha a nézetet egy tárgy tengelyéből vettük, akkor is módosíthatjuk ezzel, hiszen az előbb láttuk, hogy ekkor fizikailag nem kötődik a nézőpont a tárgyhoz, csak úgy áll be, mintha.

A **Display/Perspective Reset** alpontja a perspektívanézetet alaphelyzetbe hozza. Nincs közvetlen hatása Camera és Lightsource View esetén, mert ilyenkor a nézetet a kamera, vagy a fényforrás határozza meg.

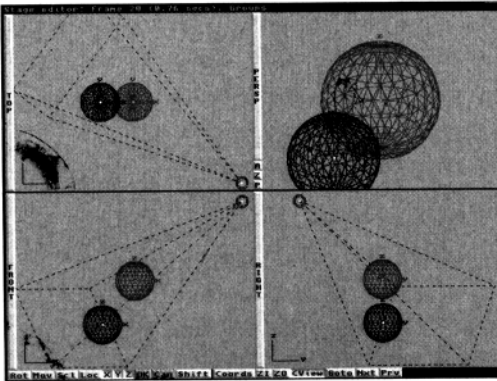
Ugyan ezen menüpont **FOV Angle** (Focus of View) alpontjával a perspektíva nézet fókuszát állíthatjuk be. A beállítás azonos azzal, amit a szerkesztőablakban a P kapcsoló aktiválása után manuálisan tettünk. A fókuszt a látószög fokban megadott értékével kell megadni. Manuálisan csak 1-150° között állíthatjuk be, de numerikusan maximum 179° látószöget adhatunk meg. A fókusz változtatásával változik a perspektivikus hatás.

A **Zoom Ratio** segítségével numerikusan adhatjuk meg a perspektívanézet zoomolásának arányát a síknézetekhez képest. Ha a **Zoom In**, vagy **Zoom Out** funkcióval változtatjuk a síknézetek nézőpont-távolságát, akkor ennek arányában a perspektívanézet is változik.

Nem közvetlenül a perspektíva nézethez tartozik, de átvittén arra is hatással van a **Display/Reset Views** menüpont, amely alaphelyzetbe állítja a szerkesztő nézeteket.

## Szerkesztési háttérképek

A Detail editornál megismert kényelmi funkció, a szerkesztési háttérképek lehetősége itt is megtalálható. A **Display/Load Backdrop** segítségével bármelyik szerkesztő negyedbe, még a perspektívába is, háttérképet tölthetünk. A kép mindig abba a nézetbe töltődik be, amelyikben utoljára klikkeltünk a bal gombbal. Nem csak önálló kép, hanem egy **IFF-Animbrush**, vagy PC-n **FLI** animáció bármelyik képkockája is betölthető. Utóbbi esetben a felrajzolás előtt egy kérdezőben kell megadni a kívánt képkocka sorszámát. A **Clear Backdrop** segítségével törölhetjük a háttérképeket.



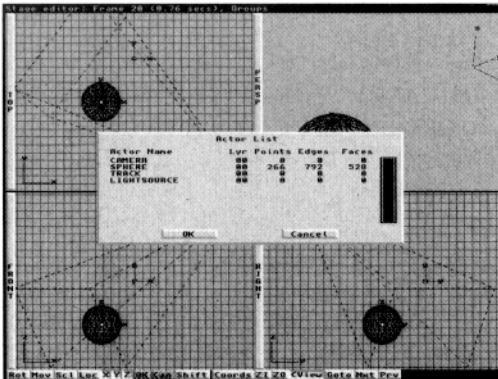
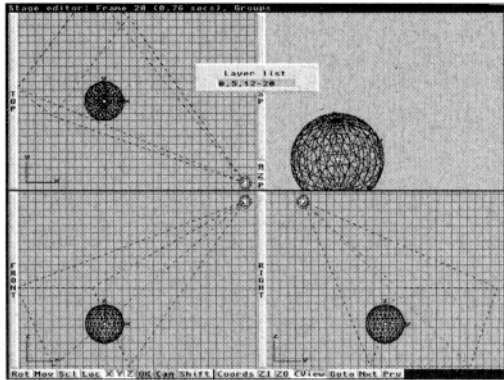
A háttérképekhez hasonló lehetőség, hogy az előzőleg megjelenített képkocka beállításait halványan, világító asztal hatását keltve megjelentethetjük háttérként, a **Display/Bluing** aktiválása után. Ez mind a három síknézetben egyszerre történik és felülbírálja a normál háttérképeket.

Az előző képkocka nem az animációs sorrendben egyel korábban szereplő képet jelenti, hanem a legutóbb megjelentettetett. Például, ha a 20. képkocka után a 2.-ra ugrunk, akkor a 20. kép lesz a háttér. A funkció használatakor sem a beállított háttérképek, sem a rácsok nem jelennek meg.

## Rétegek kezelése

Az Action editornál már szó volt arról, hogy a tárgyakat különböző logikai rétegekre tölthetjük be. Az egyes rétegeken lévő tárgyak megjelenése szabályozható. A **Display/Specify Layer** menüpontjával hívhatjuk elő azt a kérdezőt, melyben meghatározhatjuk a megjelenítendő rétegek számát. Egyszerre több réteg is megjeleníthető, a rétegek számát vesszővel elválasztva írhatjuk be. Több egymás melletti réteg a határok megadásával kötőjelezve is leírható. Például a "0,5,10-20" azt jelenti, hogy a 0-s, 5-ös, és a 10-20-as rétegeken lévő tárgyak láthatók.

Sok réteg használata esetén nehéz fejben tartani, hogy melyik rétegeken vannak a megjeleníteni kívánt tárgyak, ezért a kiválasztást két menüpont segíti. A **Display/Add Layer(s)** segítségével a már megjelenítettek mellé újabb rétegeket aktiválhatunk. A menüpontot kiválasztva megjelenik egy objektuslista, ahol kiválaszthatjuk a megjeleníteni szándékozott tárgyakat, egyszerre akár többet is, majd az OK-ra klikkelés



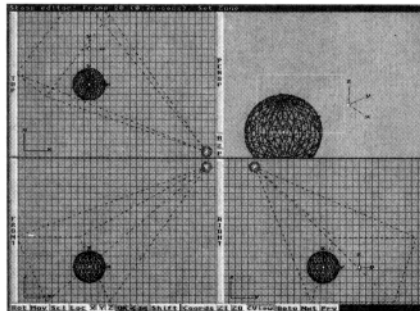
után azok a rétegek is láthatóak lesznek, amelyeken a kiválasztott tárgyak vannak. Ha több tárgy is van azonos rétegen, de csak az egyiket választottuk ki a listából, látható lesz a többi is, hiszen a funkció nem a tárgyakhoz, hanem a rétegekhez kapcsolódik.

A **Remove Layer(s)** hasonlóan működik, de ezzel a kiválasztott tárgyak rétegeinek meg nem jelenését érhetjük el. Itt is érvényes, hogy ha több tárgy tartozik azonos réteghez, de csak egyiket választottuk ki, akkor is mind rejtve marad.

### Zónák

Sokszor előfordul, hogy egy beállítás ellenőrzésére Quick Renderhez folyamodunk. Ez azonban elkészíti a teljes képet, függetlenül attól, hogy bennünket épp csak egy kis része érdekel, ezért a szükségesnél jóval több időt kell a képre való várakozással tölteni. Ezen a problémán segít, hogy a perspektíva képen kijelölhetünk egy zónát, ezután a Quick Render csak ebbe a zónába eső képrészletet készíti el, jelentősen csökkentve a várakozási időt.

A zónakijelölés igen egyszerű művelet, válaszd ki a **Display/Set Zone** menüpontot, majd a perspektíva nézetben a bal gomb lenyomásával egy befoglaló téglalappal határozd meg a zónát. Ezután az a zóna már él is, a következő QR csak az ezen belüli képrészletet számolja ki. Ha másik zónát akarsz kijelölni, válaszd ki ismét a menüpontot és határozd meg az új zónát. Zónakijelölést törölni a **Clear Zone** menüponttal lehet, ezután ismét a teljes képet készíti el a QR.



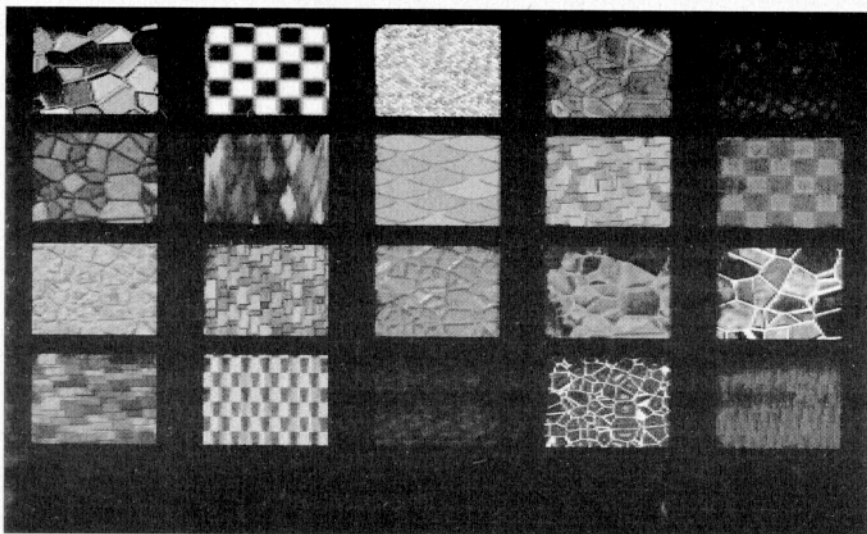


### Végül

A Stage editor utolsó újdonságával tárgyak egymáshoz viszonyított helyzetét és irányát állíthatjuk be. Tegyük fel, több tárgy van a színen, amelyből az egyiket elmozgatjuk. Miután ezt sikeresen beállítottuk, a többit hozzá kellene igazítani, hogy az egymáshoz viszonyított helyzetük pontosan olyan legyen, mint egy másik képen. Az említett funkció épp ezt végzi, egy kiválasztott tárgyhoz hozzáigazítja a többi, szintén kiválasztottat. A bázis, vagyis az a tárgy, amelyhez a többit igazítjuk, az elsőnek kiválasztottnak lesz. Miután megfelelő sorrendben kiválasztottuk a tárgyakat, jöhet az **Object/Reset Relative Position** vagy **Alignment** pontja, attól függően, hogy a pozíciót, vagy az irányt akarjuk egymáshoz állítani. Ezután meg kell adni, hogy a tárgyak mely képkockán lévő, egymáshoz viszonyított pozíciójuk, vagy irányuk legyen a beállítás alapja. Ez a kocka nem kell hogy kulcskocka legyen, vagyis nem muszály pozíció, vagy alignment felsorolójellel rendelkeznie egyetlen tárgynál sem.



# Textúrák



Az Imagine eredeti dokumentációja a textúrákat típusuk szerint csoportokra bontva tárgyalja, nem pedig ABC sorrendben. Mivel semmi nem indokolja az ettől való eltérést, én is ilyen bontásban mutatom be a azokat.

A leírás, a helyhiány miatt igen tömör formátumú, csak a 3.x-s programváltozat új textúráit mutatja be, azokat itt nem részleteztem, amelyek már a 2.0-s változatban is megvoltak, ezekről az előző kötetben volt szó. Az áttekinthetőség miatt egységes szerkezetre törekedtem, amely a következő:

**NÉV** - a textúra neve, amellyel a textures könyvtárban szerepel. Az MS-DOS korlátai miatt ez a név csak nyolc betű lehet, ezért a hosszabb nevű textúrák eredeti nevét zárójelben adom meg, ha szükséges.

## Textúrák

---

**Típus:** [Color, Filter, Reflect, Bump, Anim] - a textúra típusa, vagyis hogy a tárgy milyen paraméterére van hatással. Az anim textúrák által létrehozott mintázat animálható, hatása ismétlődő.

**Leírás** - a textúra mintázatának és használatának rövid leírása.

**Paraméterek** - a textúra paramétereinek leírása.

A részletek előtt ismerkedjünk meg néhány általános dologgal.

Bizonyos textúrák vagy felhasználják a tárgy eredeti Color, Reflect, Filter attribútumát, vagy saját beállításukkal megváltoztathatják azt. Ebben az esetben a textúra maga is rendelkezik az ennek beállítását szolgáló input mezőkkel. Ha azt szeretnénk, hogy mégis a tárgy eredeti, esetleg más textúrák és Brushok által kialakított paraméterei maradjanak meg, ne változtassa ezeket meg az adott textúra, a kérdéses paraméterhez a textúra kérdezőjében negatív számot kell írni. Ez tudatja a textúrával, hogy ezzel az attribútummal nem kell törődnie.

Némely paraméter csak bizonyos intervallumba eshet, ekkor annak a paraméternek az input mezeje mellett zárójelben megtalálható ennek az intervallumnak a leírása. Ha ezen kívül eső paramétert adunk meg a kérdezőben, az eredményt nem lehet megjósolni. Némely textúra ilyenkor egy alapértelmezés szerinti értékkel számol, de előfordulhat hibaüzenet, vagy programlefagyás is.

Előfordul, hogy a textúra a Color, Reflect és Filter értékeket is manipulálhatja, de egyszerre csak egyet. Ilyenkor a **Type Number** input mezőbe írt kapcsoló értékkel választhatjuk ki a manipulálni kívánt attribútumot. A lehetséges értékek és a jelentésük az input mező mellett van felsorolva. Általában a 0.0 jelenti a Color, az 1 a Reflect és a 2 a Filter attribútumot.

A textúrák jelentős része valamilyen fraktál zaj algoritmuson alapul. Ennek kontrolásársa két paraméter szolgál. A **Noise Magnitude** a zaj intenzitását, míg a **Noise Velocity** a frekvenciáját, vagy sebességét paraméterezi. Nagy Magnitude és kis Velocity értékkel általában a mintázat sűrű, apró lesz, míg fordított esetben nagyméretű ritkább rajzolatok alakulnak ki. Némely textúra két Noise Magnitude és Velocity paraméterrel rendelkezik, amelyek egymással interferálnak.

Az animálható textúrák esetén van egy paraméter, amely 0..1 között változtathatja az értékét, és ez szolgál időváltozóul. Statekkel változtatva ezt a paramétert, a mintázat ismétlődő változást végez. A loopolást segíti, hogy a 0.0 és az 1.0 értékeknél létrejövő mintázat azonos (vannak ez alól kivételek).

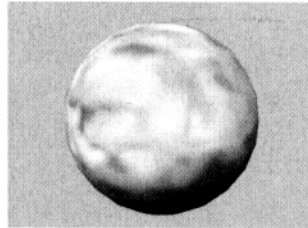
### Térkitöltő textúrák

A legnépesebb és legáltalánosasabb csoport a térkitöltő textúrák csoportja. Ebbe azok a textúrák tartoznak, amelyek a tér minden irányában azonos módon fejtik ki hatásukat.

#### AGATE

Típus: Color

Márványszerű mintázatok létrehozására szolgáló textúra. A textúra mindhárom irányban egymástól függetlenül méretezhető. A kérdezőkben három színt és négy treshold értéket lehet megadni. A treshold 0..1 között változhat és azt mutatja meg, hogy az egyes színek milyen mértékben keverednek egymással. A T1 és T2 a tárgy eredeti színe és a textúránál definiált első szín arányát, a T2 és T3 a color1 és color2, a T3 és



## Textúrák

---

T4 pedig a color2 és color3 színek keveredésének arányát mutatják.

Paraméterek:

**X, Y, Z Size** - A mintázat mérete az adott irányokban.

**Color 1, 2, 3** - A mintázat színei. Negatív értékkel csökkenthető az alkalmazott színek száma, illetve ebben az esetben azt a színt is a tárgy eredeti színe határozza meg.

### BRUSHED

Típus: Color, Bump

A drótkéfével megcsikargatott alumínium felületét utánzó mintázat. Lehetőség van az egyenetlenségek méretének és színének meghatározására.

Paraméterek:

**X, Y, Z Size** - A csikarások mérete.

**Noise 1, 2** - A sérülések szabálytalanságát befolyásoló zaj paraméterei.

**Bump Adjust** - A csikarások mélysége a tárgy eredeti méretéhez viszonyítva, értéke ezért -1..1 között lehet. A negatív érték hatására a csikarások belemélyednek a tárgyba, míg pozitív érték esetén kidudorodnak abból.

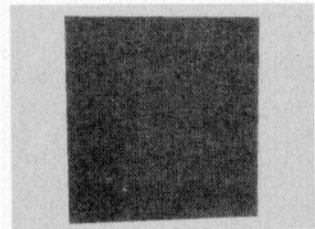
**Top Clip** - A csikarások simasága a közepükön.

**Btm Clip** - A csikarások egyenetlensége, közvetve hatással van a minták egymástól való távolságára is.

**Bump Color** - A csikarások közepének színe.

**Color Clip** - Annak mértéke, hogy a Bump Color hatása milyen messzire terjed a csikarás közepétől.

**Dispersion** - A mintázat telítettsége, sűrűsége.



## BUMPNOIZ

Típus: Bump

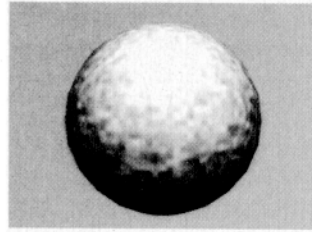
A textúra egy szabálytalan bump mintázatot generál a tárgyra, amitől annak egyenetlen lesz a felülete. A mintázat mindhárom irányban egymástól függetlenül méretezhető. Egyszerre két mintázat van a tárgyon, amelyek egymással interferálhatnak.

Paraméterek:

**X, Y, Z Size** - A mintázat mérete a három tengely irányában külön-külön.

**Bump Noise** - A mintázat zajossága, azaz egyenetlensége. Negatív érték is lehetséges.

**% Flat** - Az adott mintázat által lefedett terület aránya az eredeti mérethez képest.



## COLORNOIZ

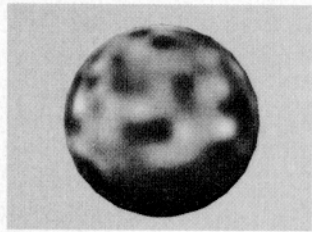
Típus: Color, Reflect, Filter

A textúra véletlen színeket visz a tárgyra, módosítva ezzel a tárgy színét, fényvisszaverő képességét és fényáteresztését. A véletlen mintázat alapja egy beállítható szín, amelynek egyes összetevőit megadott százalékos határon belül variálja a program. Külön paramétereztető, hogy az így létrejövő mintázat hány százalékban módosítsa a Color, Reflect és Filter értékeket.

Paraméterek:

**X, Y, Z Size** - A mintázat mérete az egyes irányokban.

**Color R, G, B** - A textúra bázisszíne.



## Textúrák

---

**Red, Green, Blue Vary** - Az egyes színösszetevők variálásának mértéke. Például ha a vörös összetevő 200, a Red Vary pedig 0.5, vagyis 50%, akkor a vörös színösszetevő 100..255 között fog változni. A nulla érték hatására az adott színösszetevő nem változik, míg negatív szám esetén egyáltalán nem jön létre variáció, még a többi összetevővel sem.

**Color "Value" Vary** - A szín variálásának mértéke, vagyis, hogy az előző paraméter által kialakított színek milyen mértékben hassanak a tárgyra.

**Reflect Level Vary** - A paraméter véletlen reflektiókat helyez el a tárgyon. Ha az negatív, a tárgy reflect paramétere nem változik.

**Filter Level Vary** - A paraméter véletlen átlátszóságokat helyez el a tárgyon. Ha az negatív, a tárgy filter paramétere nem változik.

**Noize Treshold** - A textúra mintázata által lefedett felület százalékos aránya. Értéke 0..1 között lehet.

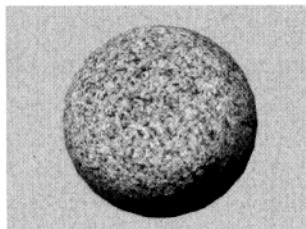
### CONCRETE

Típus: Color, Bump

A textúra betonhoz hasonló felületek létrehozását segíti. Három különböző bump map-et és egy elmosott hatású színmintázatot helyez el a tárgyon. Minden bumphoz tartozik egy szín, amelyek keverednek egymással és a tárgy eredeti színéhez, ezzel hozva létre az elmosódott színt. A bump hatás akár ki is kapcsolható, így csak a színmintázat jön létre, vagy a szín kikapcsolásával csak bump érvényesül.

Paraméterek:

**X, Y, Z Size** - A textúra mérete, amely befolyásolja a beton fő érdekességét.



**Noise** - Az egyenetlenség mértékét szabályozó fraktál zaj paraméterei.

**Color 1, 2, 3** - A három bump mintázathoz tartozó szín értékek. Ezek a színek kombinálódnak a tárgy eredeti színével. Negatív érték hatására az adott bumphöz nem fog szín tartozni.

**Bump Adjust (-1..1)** - A bump mintázat által létrehozott egyenetlenségek maximális mérete a tárgy eredeti méretéhez képest. Ha negatív, a mintázat a tárgyba befelé terjeszkedik, vagyis gödrök jönnek létre, ha pozitív, akkor kidudorodások keletkeznek.

### CONFETTI

Típus: Color, Reflect, Filter

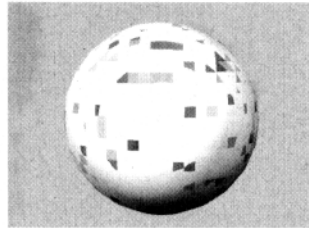
Confettiszerű színes foltokat hoz létre a tárgy felszínén. Két szín adható meg, a confettik színe e két érték között változik.

A részecskék térbeli alakok, mégpedig kockák, melyek élei párhuzamosak a textúra X, Y, Z tengelyeivel. A létrejövő mintázat ezek különböző metszeteiből áll össze. A részecskék saját reflect és filter értékkel is rendelkezhetnek, amelyek módosítják a tárgy eredeti azonos értékeit. Ezzel létrehozhatunk például opálos foltokat üveg felületen, vagy áttetsző részeket az átlátszatlan tárgyon.

Paraméterek:

**Size** - A confetti részecskék mérete.

**V1, V2** - A részecskék sűrűsége. V1-nek kisebbnek kell lenni, mint V2-nek, a kettő közötti érték a részecskék által lefedett terület aránya. Minél kisebb a két érték különbsége, annál kevesebb, ritkábban elhelyezkedő részecske keletkezik.





## Textúrák

---

**Color 1, 2** - A részecskék két határszíne. Ha csak egyszínű konfettit akarsz, mindkét értéket azonosnak kell venni. Bármelyiknél negatív számot alkalmazva, az az érték azonos lesz a tárgy eredeti színével.

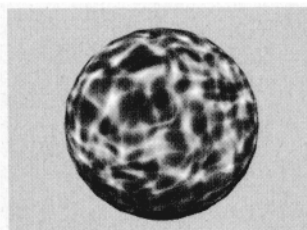
**Reflect** - A konfetti részecskék reflect értéke. Ha valamely összetevő negatív, akkor a részecskék a tárgy azonos fényvisszaverő képességűek.

**Filter** - A konfetti részecskék filter értéke. Ha valamely összetevő negatív, akkor a részecskék a tárggyal azonos fényáteresztő képességűek.

### CRUMPLED

Típus: Color, Reflect, Bump

A textúra az összegyűrt, majd kisimított papírhoz hasonló mintázatot hoz létre. A gyűrődések mérete mindhárom tengely mentén külön is állítható. A textúra szabályozhatja az egyenetlenséget, a szint, vagy mindkettőt egyszerre.



Paraméterek:

**X, Y, Z Size** - A textúra mérete a három irányban.

**Noise 1, 2** - A gyűrődéseket létrehozó fraktál zaj paraméterei. A magnitude a gyűrődések méretét, még a velocity a sűrűségét szabályozza.

**Bump Adj** - A gyűrődések magassága, vagy negatív érték esetén a mélysége. Ha nulla, nem jön létre egyenetlenség.

**Top/Bottom Clip** - A gyűrődések tetejének és aljának kisimultsága.

**Bump Color** - A gyűrődések közepének színe. Maga a gyűrődés a tárgy eredeti színe és e között az érték között színeződik. Ha valamely összetevőnek negatív az értéke, nem színeződik a bump. Ha

alkalmazásra kerül a színezés, akkor az csökkenti az eredeti reflect értéket is azon a területen, ahol felhasználásra kerül.

**Color Clip** - A bump színezésének mértéke. Kisebb érték esetén csak a gyűrődés közepéhez közeli területek színeződnek, míg nagyobb érték esetén a távolabbiak is.

**Dispersion** - A gyűrődések elterjedésének mértéke. Minél nagyobb, annál gyűröttebb a tárgy.

### DINOSKIN

Típus: Color, Bump

Rücskös dinoszaurusz bőrhöz hasonló mintázatot eredményező textúra. Két zaj paraméterrel rendelkezik, az első a rücskök létrehozásáért felelős, a másik a létrejött rücskök szélének egyenetlenségét szabályozza. A rücskök tetejének színe külön is megadható.

Paraméterek:

**X, Y, Z Size** - Az egyenetlenségek általános mérete.

**Noise 1** - A mintázat alap egyenetlenségének paraméterei.

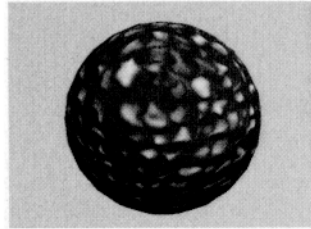
**Noise 2** - Az előző noise érték által létrehozott bump mintázat széleinek egyenetlensége.

**Bump Adj** - A rücskök magassága, vagy mélysége.

**Top/Bottom Clip** - A mintázat tetejének és aljának megnyirbáltsága. Maximális érték mindkét esetben 1.73.

**Bump Color** - A rücskök tetejének a színe. Negatív értékkel kikapcsolható a külön szín.

**Color Clip** - A rücskök színezésének mértéke, vagyis, hogy a Bump Color mekkora területen alkalmazódjon.

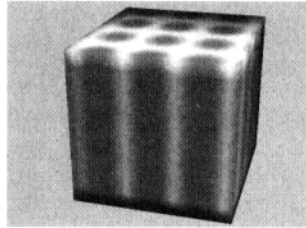
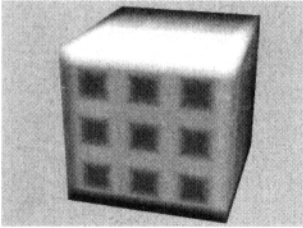


# Textúrák

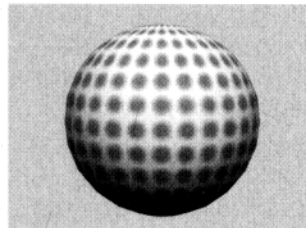
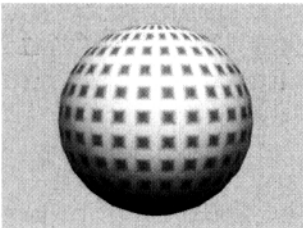
---

## DITHERED SHAPE család

### DITHRECT, DITHCIRC,



### RDDITREC, RDDITCIR



Típus: Color, Reflect, Filter

A család textúrái négyzeteket, vagy köröket rajzolnak a tárgy felületére. A mintázat szélének áttűnése szabályozható, így akár éles, akár lágy kontúrok kialakíthatók. A mintázat ditherelése noise paraméterrel befolyásolható.

Általános paraméterek:

**Size Adjust** - Az egyes minták mérete a textúra méretéhez képest. A textúra általános mérete a textúra tengelyének méretével szabályozható.

**Blending Width** - A mintázat lágy szélátmenetének szélessége a minta teljes méretéhez képest.

**Noise 1, 2** - A blending szabálytalanságát befolyásoló zaj paraméterei.

**Intensity Adjust** - A textúra mintázatának fedőképessége. Alacsonyabb érték esetén kevésbé fedik a minták a tárgy eredeti attribútumát, az átlátszik. Folyamatosan változtatva, a mintázat fadelhető.

**Color/Reflect/Filter** - A mintázat attribútumai.

**Arc Division Adjust** - A Rd... nevű textúráknál található paraméter, a kerületen elhelyezkedő minták száma.

### DIRT

Típus: Color, Reflect, Filter

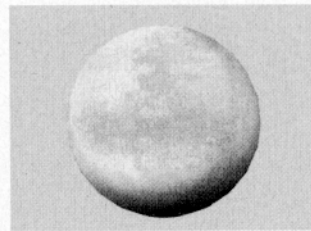
A textúra mindössze bepiszkítja a tárgy felületét, vagyis olyan hatást kelt, mintha az piszkos volna. Nem csak a színt lehet ilyen módon megváltoztatni, hanem a reflect és filter értékeket is.

Paraméterek;

**Noise 1, 2** - A piszkosságot kialakító zaj paraméterei.

**Dirt Intensity** - A piszok erőssége.

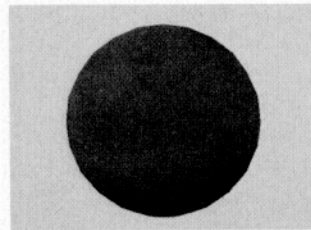
**Dirt Color, Reflect, Filter** - A piszok színe, reflectje és filtere. Negatív értékkel az adott paraméterre gyakorolt hatás kikapcsolható.



### DRTPAINT

Típus: Color, Reflect, Filter

Ez a textúra hasonlóan működik a Dirthez, kivéve, hogy a mintázat intenzitása a tárgy már meglévő attribútumaihoz igazodik. A piszok kialakuló színét egy input mezőben megadott maximum érték és a tárgy valamely



## Textúrák

---

attribútumának vörös intenzitása határozza meg. Együtt használva egy vörös brush mappal, vagy vörös értéket manipuláló textúrával a piszok megjelenése lekorlátozható a tárgy bizonyos részeire. A textúra tetszés szerint kulcsolható a Color, Reflect vagy Filter attribútumhoz. Ha a textúrát a Color attribútumhoz kulcsoljuk, az eredeti szín helyett egy megadott homogén szín fog megjelenni a tárgyon, míg ha a Reflect, vagy Filter attribútumhoz, azok aktuális beállítása figyelmen kívül lesz hagyva. A szennyezés ki is emelkedhet a tárgyból, egy bump paraméter segítségével. A textúra méretének és elhelyezkedésének beállítását a textúra tengelyével végezhetjük.

Paraméterek:

**Noise 1, 2** - A piszkosságot kialakító zaj paraméterei.

**Max Dirt Intensity (0..1)** - A piszok maximális intenzitása azon a területen, ahol a meghatározó attribútum vörös összetevője maximális.

**Dirt Color R, G, B** - A piszok színe.

**Object Color R, G, B** - A tárgy színe abban az esetben, ha a Dirt a Color attribútumhoz van kapcsolva.

**Bump Adjust** - A szennyeződés egyenetlensége.

**Key off** - Kapcsoló annak meghatározására, hogy az object mely attribútuma legyen a szennyeződés mértékének alapja.

### EASYWOOD

Típus: Color

A textúra úgy alakítja a tárgy színeit, mintha az fából lenne kivágva. A fa évgyűrűi a Z tengely mint középpont körül ala-



kulnak ki. A gyűrűk színe a kérdezőből állítható, míg a közöttük lévő területet a tárgy eredeti színe alakítja ki.

Paraméterek:

**X, Y, Z Size** - A gyűrűk mérete, mindhárom irányban függetlenül változtatható.

**Noise (0..1)** - A gyűrűk X-Y irányú egyenetlenségének mértéke, vagyis a gyűrűszél csipkézettsége.

**Ring Width (0..1)** - A gyűrűk szélessége a köztük lévő helyhez viszonyítva. Ha ez az érték 1, a gyűrűk teljesen összeérnek, Ha 0, nem is lesznek évgyűrűk.

**Ring Vibration** - Az évgyűrűk Z tengely mentén létrejövő hullámossága.

**Color, Reflect, Filter** - A fa gyűrűinek attribútumai.

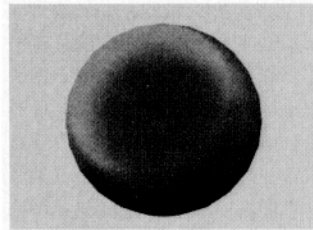
### FAKELY

Típus: Color, Reflect, Filter

A textúra hasonló hatást eredményez, mint a napszemüvegen létrejövő fényvisszaverődés. Egy színátmenet jön létre két szín között. Az egyik szín mindig a kamerával szemben, a tárgy közepén jelenik meg, a másik pedig ettől 90°-ra a tárgy oldalán körben. A két szín súlyozható, vagyis megadható, hogy melyik milyen erősen érvényesüljön.

Paraméterek:

**Type Number** - Kapcsoló annak kiválasztására, hogy a tárgy mely attribútumát manipulálja a textúra.



## Textúrák

---

**Change Thresh (0..1)** - A két szín erősségének aránya. Ha értéke 0, a tárgy teljes egészében a front színben tűnik fel, ha 1.0, akkor teljesen az oldal színben. A 0.5 érték hatására a két szín egyenlő mértékben kerül alkalmazásra.

**Side R, G, B** - A tárgy oldalának a színe.

**Front R, G, B** - A tárgy kamera felé eső oldalának színe.

### FROGSKIN

Típus: Color, Reflect, Filter

A textúra egy foltos, rücskös, pettyes felületet képez, amely hasonló a béka bőréhez. A textúra csak a foltok és pettyek színét módosítja, az ezek közötti rész színét a tárgy eredeti attribútuma határozza meg. A foltok méretét a textúra tengelyének méretével lehet szabályozni, míg az azokon lévő pöttyökét egy kérdezőben adhatjuk meg. A foltok és a pöttyök színe külön beállítható, noise paraméterrel változtatható az alakjuk ezen kívül, a pöttyökhöz bump is tartozhat.

Paraméterek:

**X, Y, Z Size** - A pöttyök mérete a három irányban egymástól függetlenül. A foltok méretét a textúra tengelyének mérete határozza meg.

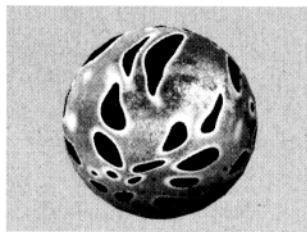
**Noise 1, 2** - A foltok és a pöttyök kialakulásáért felelős fraktál zaj paraméterei.

**Bump Adjust** - A pöttyök kimagasodásának mértéke. Ha értéke negatív, nem kidudorodás, hanem bemélyedés jön létre.

**Spot Size Adjust** - A pöttyök által lefedett terület-rész aránya.

**Spot Color R, G, B** - A pöttyök színe.

**Speckle Color R, G, B** - A foltok színe.



## JERSEY

Típus: Color, Reflect, Filter

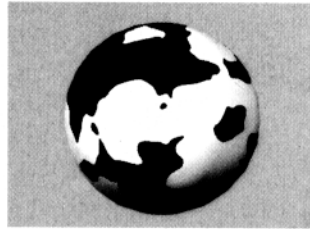
Véletlen elhelyezkedésű, alakú és méretű foltokat helyez a tárgyra, amitől az hasonlóan fog kinézni, mint a tarka tehén. A foltok alakját némiképp befolyásolhatjuk a textúra méretét változtatva. Szabályozható a foltok által lefedett terület aránya, két Noise paraméterrel pedig elhelyezkedésének és egyenetlenségének mértéke.

Paraméterek:

**X, Y, Z Size** - A textúra mérete.

**Noise 1, 2** - A textúra elhelyezkedésének és a szélének szabálytalansága.

**Color, Reflect, Filter** - A foltok attribútumai.



## LEATHER

Típus: Bump

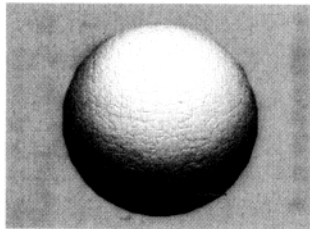
A Leather csak a felület egyenetlenségét manipulálja, bőrszerű, ráncos felület hatását keltve. A ráncok szélessége és magassága szabályozható.

Paraméterek:

**X, Y, Z Size** - A ráncok méretének értékei. Egy tengely mentén megnövelve a ráncokat, azok orientálhatók.

**V1, V2** - A ráncok szélességének alsó és felső értékei. V1-nek kisebbnek kell lenni, mint V2-nek.

**Bump Adjust** - A redők mélysége/magassága.





### MARBLE

Típus: Color, Reflect, Filter

A textúra egy megadott színt kever a tárgy eredeti színéhez, különböző mértékű elmosásokat alkalmazva, ezzel márványszerű hatást hozva létre. Az erezet a textúra tengelyén keresztül méretezhető.

Paraméterek:

**Noise 1, 2** - A mintázat szabálytalanságának paraméterei.

**Intensity** - Az erezet színének érvényesülése a tárgy eredeti színén. Kisebb értéknél halványabb mintázat jön létre.

**Grain Color** - A márvány erezetének színe.

**Threshold** - A márvány erezete által lefedett terület aránya a tárgy méretéhez képest.



### MONSTER

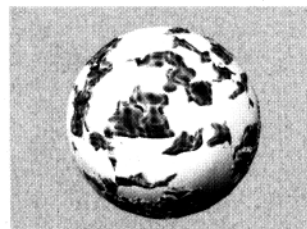
Típus: Color, Bump

A textúrával véres darált húshoz hasonló, romlásnak indult felületek hozhatók létre. A hatást egymásra helyezett bump mintázatok hozzák létre, nagy csomókon elhelyezkedő kisebb dudorok, amelyeket két színnel lehet színeezni. A mintázat eloszlása is szabályozható.

Paraméterek:

**X, Y, Z Size** - A mintázat átlagos mérete.

**Noise 1, 2** - Két fraktál zaj paraméter az egyenetlenségek kialakítására. A magnitude szabályozza, hogy mekkora mértékben változzon meg a textúra alakja, míg a velocity az alak egyenetlenségével van összefüggésben. A Noise 1 az elsődleges egyenet-



lenségért felelős, jó ha nagy magnitúdót és kis velocity értéket állítunk hozzá, így nagy, aránylag szabályos cafatok jönnek létre. A Noise 2 a másodlagos egyenetlenséget szabályozza, ehhez kis amplitúdót és nagy velocity-t adva érhetjük el a legjobb darált hús felületet.

**Bump Adjust** - Az egyenetlenségek magassága/mélysége.

**Bottom Clip (0..0.71)** - A csomók és a tárgy találkozási helyének paramétere. Minél kisebb ez az érték, annál finomabb az átmenet a tárgy és a csomó között.

**Big Bump Color** - A csomók színe.

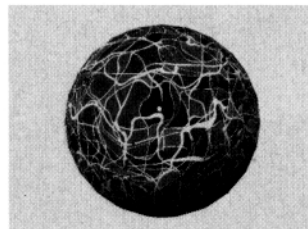
**Small Bump Color** - A csomókon lévő dudorok színe. Utóbbi két paraméternél negatív értéket használva az adott egyenetlenségnél nem változik a tárgy eredeti színe.

**Dispersion** - A csomók által lefedett terület aránya. Ha értéke 1.0, az egész felületen megjelennek a csomók, 0.5 értéknél csak a tárgy felületének felét borítják be.

### MOSAIC

Típus: Color, Reflect, Filter, Bump

A textúra segítségével mozaikból, törött színes csempékből kialakított felületet utánozhatunk. A színezéshez három színt használhatunk, ebből kettőt a mozaikdarabok szélső színeiként, egyet pedig a köztük lévő anyag színeiként.



## Textúrák

---

Paraméterek:

**Scale Adjust** - A csempék mérete. Csak egy ilyen paraméter van, ezért nem lehet irányonként függetlenül beállítani a csempék méretét.

**Color 1, 2** - A csempék szélső színei. Az egyes darabok ezek közötti véletlen színeket fognak felvenni. Ha valamelyiknek negatív számot adunk meg, az a szín azonos lesz a tárgy már meglévő színével.

**Bump Adjust** - A csempék nem csak egy síkban lehetnek, hanem szabálytalanul ki is lóghatnak abból, mintha nem szakiparos rakta volna le azokat.

**Grout Width** - A csempék között kötőanyag is lehet, ez a paraméter határozza meg annak a szélességét a teljes felülethez képest. Ha nincs szükség illesztékre, értékének 0.0-t kell adni. Az 1.0-es szélesség azt eredményezi, hogy nem is lesz csempe, csak kötőanyag.

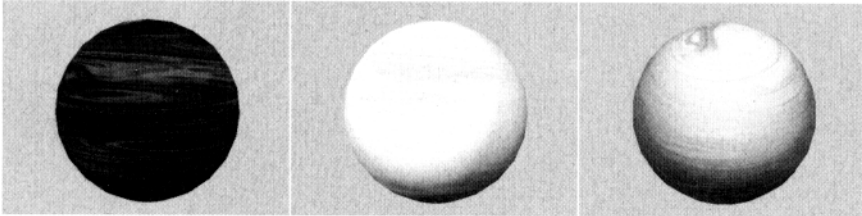
**Grout Color** - A csempék közötti anyag színe.

**Filter/Reflect** - A csempék átlátszóak, fényvisszaverők is lehetnek, ez a paraméter határozza a mértékét.

**Filter/Reflect Vary** - Az előző paraméterrel beállított Filter és Reflect értékeket véletlen algoritmus-sal módosíthatjuk is, a maximális módosítás mértéke ez a paraméter.

## Noiz2 család

### CLRNOIZ2, REFNOIZ2, FILNOIZ2



A Noiz2 család tagjai véletlen színeket visznek a tárgy felületére. Az eredményt három szín kombinálása adja. A textúra ezek között, mint határszínek között hoz létre új színeket véletlen eloszlásban. Az eredmény olyan, mintha különböző színű festékeket folyasztottunk volna a tárgyra. A három textúra azonos algoritmust használ, a különbség közöttük, hogy a Color, Reflect és Filter paramétereket manipulálják.

Paraméterek:

- X, Y, Z Scale** - A festékfoltok átlagos mérete. Az egyik irányban jelentősen emelve a méretet, olyan hatást érhetünk el, mintha a festék megfolyt volna.
- Noise 1, 2** - A mintázat össze-visszaságának mértéke. A Magnitude a foltok alakját, a Velocity pedig a színük tarkaságát szabályozza.
- Color/Reflect/Filter 1, 2, 3** - Három szín alapú paraméter, minden textúránál a neki megfelelő attribútumtípus található.

### OLDBRICK

Típus: Color

Ez a textúra hasonló a Brick-hez, segítségével téglamintázatot hozhatunk létre, de ezek a téglák nem szabályosak, hanem piszkosak, repedezett szélűek, sőt méretük sem egyforma, mint egy régi téglafal. A téglák egyenetlenségét bump paraméterrel is szabályozhatjuk. Az öreg, piszkos hatás kialakításához a téglának két határszint adhatunk meg, az aktuális színe e két érték között véletlenszerűen alakul. A téglák között kötőanyag van, melynek színe a tárgy eredeti színével azonos.

Paraméterek:

**X/Y Size** - A téglák mérete X és Y irányban. A Z méret a textúra tengelyétől függ.

**Bump Adjust** - A téglák rücskösségének magassága, mélysége.

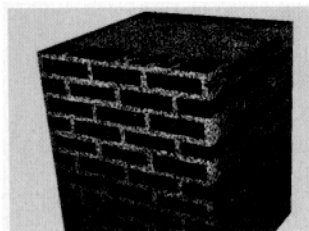
**Brick Edge Width** - A téglák élének szélessége.

**Shape/Size Disturb** - A téglák alakjának és méretének változatossága.

**Brick/Mortal Dirt** - A téglák és a közöttük lévő kötőanyag felületének piszkossága.

**Stagger Adjust** - Az egyes téglasorok eltolása egymáshoz képest.

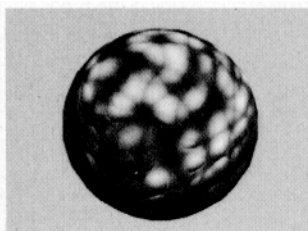
**Brick Color 1/2** - A téglák két bázisszíne, a valós színek e két határ között véletlenszerűen alakul.



### PEBBLED

Típus: Color, Reflect, Bump

A textúra olyan hatást eredményez, mintha a tárgy kavicsokból állna. A hatást rengeteg egymáshoz szorosan illeszkedő, vagy egymást fedő gömbszerű dudorral éri el, amelyek között részt színezhetsi is.



Paraméterek:

**X, Y, Z Size** - A dudorok mérete, a három tengely mentén egymástól függetlenül.

**Noise 1, 2** - A dudorokat kialakító zaj paraméterei.

**Bump Adj** - A dudorok magassága (vagy mélysége, ha negatív).

**Top Clip** - A dudorok felső részének kivágása.

**Bttm Clip** - A dudorok aljának egymástól való távolsága.

**Bump Color** - A dudorok közötti árkok színe. Negatív érték használata esetén nem színeződnek el ezek a felületrészek.

**Color Clip** - Annak mértéke, hogy a Bump Color milyen széles területet fedjen be a két érintkező dudor között. Minél nagyobb ez az érték, annál inkább eluralkodik a Bump Color.

**Dispersion** - A dudorok által lefedett terület aránya a tárgy egész felületéhez képest. ha 0, nem jön létre egyenetlenség, ha 1.0, a teljes felületet egyenetlenné válik.

### PEENED

Típus: Color, Reflect, Bump

Ezzel a textúrával olyan felületeket hozhatunk létre, mintha azok össze volnának kalapáccsal verve. A hatást egymással érintkező, vagy átfedő körkörös horpadások hozzák létre, amelyeket színezhettek is.

Paraméterek:

**X, Y, Z Size** - A horpadások mérete, a három tengely mentén egymástól függetlenül.

**Noise 1, 2** - A horpadásokat kialakító zaj paraméterei.

**Bump Adj** - A horpadások mélysége (vagy magassága ha pozitív).

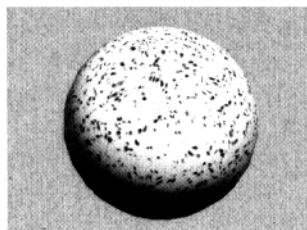
**Top Clip** - A horpadások közepének simasága.

**Btm Clip** - A horpadások közötti részek illeszkedése.

**Bump Color** - A horpadások színe. Negatív érték használata esetén nem színeződnek el ezek a felületrészek. A Bump Color a horpadás közepén érvényesül és attól kifelé gyengül, teret adva a tárgy színének.

**Color Clip** - A bump színének erőteljessége. Minél nagyobb ez az érték, annál nagyobb területet terít be a Bump Color.

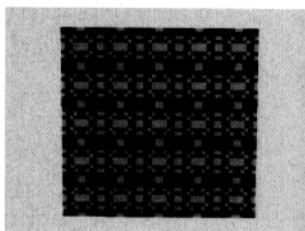
**Dispersion** - A horpadások által lefedett terület aránya a tárgy egész felületéhez képest. ha 0, nem jön létre egyenetlenség, ha 1.0, a teljes felületet egyetlenné válik.



**QUILT**

Típus: Color

A textúra véletlenszerűen elhelyezkedő, ismétlődő négyzetekből álló mintázatot rajzol a tárgyra. A blokkok három szín valamelyikében jelenhetnek meg, méretük, számuk, intenzitásuk szabályozható.



Paraméterek:

**Random Seed** - A mintázat egyénibbé tételét szolgáló véletlen szám.

**Pattern Pieces** - A blokkok száma az ismétlődő patternen belül.

**Colors 1, 2, 3** - A három szín a blokkok számára.

**Intensity Adjust** - A blokkok intenzitása, az eredeti mintázatot lefedő képességük.

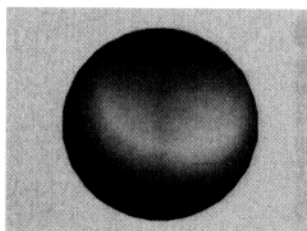
**RAINBOW**

Típus: Color

A textúra irányfüggően viselkedik a tárgyra. Három színt adhatunk meg, ezek a három tengely mentén kerülnek a tárgyra. Az egyes színek intenzitása a tengelyponttól való távolsággal arányos. A színek mellett lehetőség van a relatív fényvisszaverő és fényáteresztő képesség módosítására is. A színátmenet hossza, vagyis az a távolság, amely után az adott tengelynek megfelelő szín eluralkodik a tárgyon a textúra tengelyének méretével szabályozható.

Paraméterek:

**X, Y, Z Color** - A három tengely mentén felkerülő színek.





## Textúrák

---

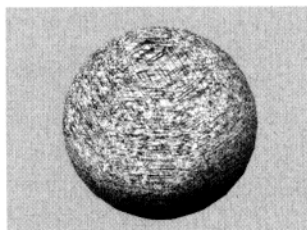
**X, Y, Z Reflect (0..1)** - Az eredeti Reflect érték módosításának maximális mértéke. Az aktuális módosítás szintén a tengelyponttól való távolság függvénye. Negatív érték beírásával a módosítás kikapcsolható.

**X, Y, Z Filter (0..1)** - Az eredeti Filter érték maximális módosításának mértéke. Az aktuális érték szintén a tengelyponttól való távolság függvénye. Negatív érték beírásával a módosítás kikapcsolható.

### SCRATCH

Típus: Color, Reflect, Bump

A textúra véletlen eloszlású karcolásokat és horpadásokat hoz létre a tárgyon, ütött-kopott kinézetet adva neki. A hatást különböző hosszúságú csikarások, bevágások és horpadások elhelyezésével hozza létre, amelyek egymást átfedhetik és interferálhatnak. A sérülések színezésére külön színt is használhatunk, még életszerűbbé téve a tárgy ócskaságát.



Paraméterek:

**X, Y, Z Size** - A textúra mérete a három tengely mentén külön-külön. A legjobb eredményt az adja, ha az egyik irányban a méret egy nagyságrenddel nagyobb, mint a másik irányban. A harmadik tengely mentén még ennél is kisebb méretet kell megadni, mint ahogy az az alapbeállításnál is látható.

**Noise 1, 2** - A sérülések nyomait kialakító fraktál zaj paraméterei. A magnitude érték jelzi, hogy milyen mértékben módosítsa a tárgy alakját a sérülés, a velocity, pedig magának a sérülés formájára utal.

**Bump Adjust** - A sérülések mélysége, vagy pozitív érték esetén a magassága.

**Top Clip** - A sérülések közepének simasága.

**Bottom Clip** - A sérülések közötti felületek simasága, illeszkedése a sérülésekhez.

**Bump Color** - A sérülések közepének színe. Bármely színösszetevőnek negatív értéket adva, nem kerül külön szín alkalmazásra.

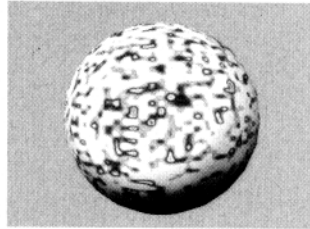
**Color Clip (1..1.73)** - Annak mértéke, hogy a sérülés közepére kerülő Bump Color milyen mértékben uralkodjon el a sérülésen. Minél nagyobb ez az érték, középről kiindulva annál nagyobb sugarú területen színezi át a sérülést a textúra.

**Dispersion (0..1)**- A sérülések által lefedett felület aránya az egész felülethez képest. Ha értéke 0, nem módosítja a tárgyat a textúra.

### SLOTCH

Típus: Color

Ennek a textúrának a működése igen bonyolult. Több réteget definiálhatunk, amelyeken belül a tárgy eredeti színe módosul a fekete szín felé a megadott arányban. A szemléletesség kedvéért vegyük a tárgy felületét 100%-nak. Tegyük fel, hogy  $T1=20$ ,  $T2=30$ ,  $T3=40$ , még ha kezdetben nem is tudjuk mi az. Az első  $T1$  %-nyi részen, tehát az első 20% felületrészen a színek nem módosulnak, az ezt követő  $T1-T2$  részen, azaz a 20 és 30% közötti felületrészen a jelen lévő színek  $F1$  arányában közelítenek a feketéhez. A  $T2-T3$  felületrészen, azaz 30-40% közötti területen  $F2$  arányában közelítenek a feketéhez. Összesen 8 felületarányt határozhatunk meg,  $T8$ -hoz  $F7$  tartozik. Fentiekből látszik, hogy  $T2$ -nek nagyobb-



## Textúrák

---

nak kell lenni T1-nél de kisebbnek T3-nál. Ez az összefüggés érvényes a többi T paraméterre is. F értékei szabadon változhatnak 0-1 között. A valóságban ezek a százalékosan meghatározott sávok nem folyamatosan, hanem véletlen eloszlásban szerepelnek a tárgy felszínén, foltos kinézetet adva annak.

Paraméterek:

**Size** - A foltok mérete.

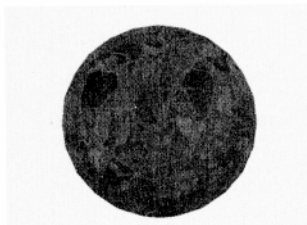
**F1..F7** - A módosítások aránya. Ha Fx értéke 1.0, teljes mértékben a tárgy eredeti színe érvényesül, ha 0.0, az adott felületrészen fekete lesz a tárgy. A 0.5 érték azt jelenti, hogy az eredeti szín és a fekete 50%-os keveréke lesz a tárgy színe a jelzett felületrészen.

**T1..T8** - Az egyes felületrészek arányai.

### STAINGLS

Típus: Color, Filter, Bump

A textúra színes berakott üveg felületet hoz létre. Ehhez véletlen eloszlású, alakú és színű lapokat használ, amelyek közé egy megadott színű kötőanyag réteget is helyezhetünk. A textúra a rajta áthaladó, árnyékvetés tulajdonsággal rendelkező fényforrások fényének színét módosítani tudja, színes fényárnyékot létrehozva ezzel. A lapok illeszkedésének egyenletlenséget is adhatunk egy bump paraméter segítségével. A textúra két színt használ a darabok színének meghatározásához. Az egyes lapok e két szín közötti értékeket vesznek fel. A kötőanyag vastagsága és színe külön szabályozható.



Paraméterek:

**Scale Adjust** - Az üvegcserепek átlagos mérete. A megadott méret a cserép egészére vonatkozik, nem méretezhetjük tengelyirányonként külön-külön.

**Color 1, 2** - Az üvegdarabok két szélső színe. Az egyes szeletek e két határszín közötti véletlen színeket vesznek fel.

**Bump Adjust** - Az üvegfelületek egyenetlensége.

**Leading Width** - Az üvegcserепek közötti kötőanyag vastagságának aránya a textúra által lefedett felülethez képest. Ha értéke 0.0, nem lesz kötőanyag, ha 1.0, nem lesznek cserепek. Szokásos értéke 0.1 körüli.

**Leading Color** - A kötőanyag színe. Bármely színösszetevőnek negatív értéket adva a kötőanyag színét a tárgy eredeti (esetleg magasabb prioritású textúrák és brushok által már módosított) színe adja.

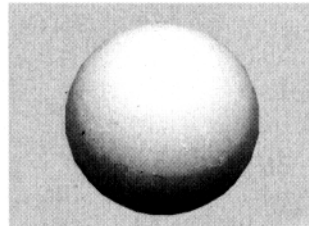
**Bump Velocity** - Az üvegen lévő bump map mérete.

**Transparency** - Az egyes üvegcserепek átlagos átlátszósága.

### STATUE

Típus: Color, Bump

Ez a textúra véletlen alakú és elhelyezkedésű bemélyedések megjelenítésével kopott kőszobor látszatát próbálja kelteni. Az egyenetlenségeken kívül ehhez egy külön szín is alkalmazható. A textúra felhasználható még a szín és az egyenetlenségek méretének megfelelő beállításával borszerű felületek létrehozásához is. A min-tázathoz két noise paramétert használhatunk.



## Textúrák

---

Paraméterek:

**X, Y, Z Size** - A textúra mérete, amelyet egy irányban jelentősen megnövelve, a mintázat abba az irányba orientálható.

**Noise 1, 2** - A mintázat változatosságának mértéke.

**Bump Adjust** - A mintázat besüllyedésének, kiemelkedésének mértéke.

**Top Clip** - Az egyenetlenségek közepének simasága.

**Bottom Clip** - Két egyenetlenség közötti rész simasága és illeszkedése az egyenetlenségekhez.

**Bump Color** - A deformált rész közepére kerülő szín. Bármelyik színösszetevőnek negatív értéket adva, nem fog külön színeződni ez a terület.

**Color Clip** - A Bump Color színének területe. Minél nagyobb ez az érték, az egyenetlenség közepétől számítva annál nagyobb sugarú területet fed be Bump Color.

**Dispersion** - A mintázat által lefedett terület aránya a textúra által fedett összes területhez képest.

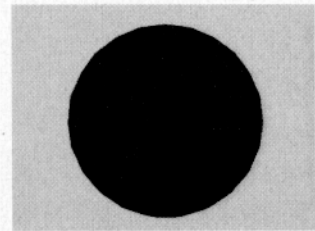
### TERRA

Típus: Color, Bump

A Terra textúra két független bump funkció segítségével olyan felületet alakít ki, mint egy bolygófelszín. Mindkét bump funkcióhoz tartozik egy szín és egy blending érték, amely azt mutatja meg, hogy az adott szín milyen erősen érvényesüljön. A mintázat mérete a textúra tengelyével állítható.

Paraméterek:

**Noise 1, 2** - A mintázat gyakorisága és alakja, két fraktál noise alapú függvényben kifejezve.



**Color 1 R, G, B** - Az elsődleges bump map által fedett terület színe. Ha a blending be van kapcsolva, ez a szín a bump közepén dominál, a széle felé az eredeti, vagy a másik bump színe veszi át a hangsúlyt.

**Color 1 Bump (-1..1)** - Az elsődleges bump egyenetlenségének mértéke.

**Blend Color 1? (0/1)** - A blending ki-be kapcsolását végző paraméter. Ha értéke 0.0, nincs blending, ha 1.0, a blending be van kapcsolva.

**Color 2 R, G, B** - A másodlagos bump színe. Ha a blending be van kapcsolva, ez a szín a bump közepén dominál, a széle felé az eredeti, vagy a másik bump színe veszi át a hangsúlyt.

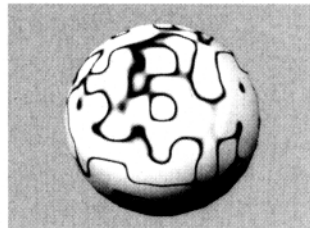
**Color 2 Bump (-1..1)** - A másodlagos bump egyenetlenségének mértéke.

**Blend Color 2? (0/1)** - A blending ki-be kapcsolását végző paraméter. Ha értéke 0.0 nincs blending, ha 1, a blending be van kapcsolva.

### WORMVEIN

Típus: Color, Reflect, Filter, Bump

A textúrával féregjáratokat hozhatunk létre a tárgy felszínén. Ezek úgy néznek ki, mintha a felszín alatt férgek mászkáltak volna, amitől a felület feldűródött. A hatás eléréséhez a bump paraméter mellett Color, Reflect és Filter tulajdonságokat is vihetünk a féregjáratokra. A textúra a három tengely mentén függetlenül méretezhető, az egyik irányban lényegesen nagyobb értéket adva, olyan hatást válthatunk ki, mintha a férgek egy irányba tartottak volna. A féregjáratok szélességét külön is beállíthatjuk.



## Textúrák

---

Paraméterek:

**X, Y, Z Size** - A textúra méretének paraméterei.

**V1, V2** A féregjáratok szélességének paraméterei. A V1-V2 közé eső felületrész-százalék lesz a féregjárat helye. Ez úgy értendő, mint a Splotch textúránál a Tx paraméterek.

**Color/Reflect/Filter** - A féregjárat attribútumai.

**Bump Adjust** - A féregjárat kimagasodása (vagy besülyedése, ha negatív).

### VRINKLE

Típus: Color, Bump

Ez egy bump noise típusú textúra, amely kusza rajzolattal hálózza be a tárgyat, ráncos kinézetűvé téve azt. A bump akár színezhető is, két határszint adhatunk meg, a mindenkori aktuális szín ezek közötti véletlen érték lesz.

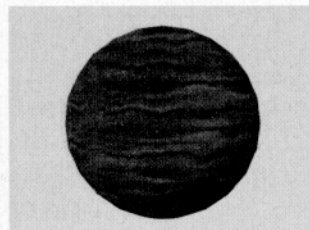
Paraméterek:

**X, Y, Z Size** - A ráncok átlagos mérete.

**Noise 1, 2** - A ráncokat kialakító két fraktál zaj paraméterei.

**Bump Adjust** - A ráncok magasságának (vagy mélységének) mértéke.

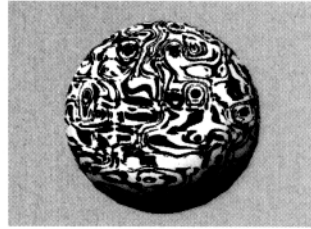
**Color 1, 2** - A két határszín a ráncok színezéséhez. Negatív érték esetén az a szín azonos lesz a tárgy eredeti színével. Mindkét szín valamely színösszetevőjéhez negatív értéket írva nem színeződnek a ráncok.



## ZOOLÓ

Típus: Color, Reflect, Filter, Bump

A textúra hosszabb-rövidebb szalag alakú, véletlen elhelyezkedésű foltokat helyez a tárgyra. A foltok mérete a három tengely irányában szabadon változtatható, egy irányban erősen megnyújtva, a szalagok orientálhatók.



Paraméterek:

**X, Y, Z Scale** - A foltok átlagos mérete.

**Disturbance** - A mintázat bump értékének változossága. Minél nagyobb ez az érték, annál inkább gyűröttebb, kuszább hatású szalagok rajzolódnak ki.

**Dispersion** - A foltok által lefedett terület aránya.

**Percent Colored** - Az egyes szalagok mérete. Minél kisebb ez az érték, annál rövidebb, kisebb darabokból fog felépülni a mintázat.

**Stripe Color** - A szalagok színe. Valamely színösszetevőnek negatív értéket adva a mintázat nem vesz fel külön színt.

**Stripe Reflect** - A szalagok által létrehozott reflect módosítás maximális mértéke. Negatív értékkel kikapcsolható.

**Stripe Filter** - A szalagok által létrehozott filter módosítás maximális mértéke. Negatív értékkel kikapcsolható a módosítás.

**Angle Vary** - A szalagok girbe-gurbasága. Minél nagyobb ez az érték, annál kuszább lesz a mintázat vonalvezetése.



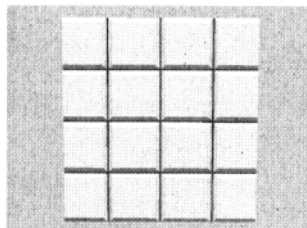
### 2D textúrák

A második legnagyobb textúra csoport olyan mintázatokot tartalmaz, amelyek csak két dimenzióban fejtik ki hatásukat. Általában ez az X-Y sík, de vannak kivételek. A működés síkját minden textúránál külön megemlítem.

#### BATHTILE

Típus: Color, Bump

A textúra az X-Y síkon hoz létre csempemintázatot, tehát ahhoz hogy hatása jelentkezzen, a Z tengelynek a tárgyból kifelé kell mutatni. A textúrával sarkított élű csempemintázatot hozhatunk létre. A csempék attribútuma azonos a tárgyével, de lehetőség van egy más színű, szabályozható szélességű kötőréteget tenni a csempék közé. Egy eltolási paraméterrel az egyes csempesorok egymáshoz képest eltolhatók, téglaszerű mintázatot létrehozva ezzel. Arra is van mód, hogy a textúrát minden második csempén kikapcsoljuk, saktábla hatást érve el.



Paraméterek:

**X, Y, Size** - A csempék mérete. A két értéknek nem kell azonosnak lenni, tehát a csempék téglalap alakúak is lehetnek.

**Bevel Width** - A csempék élettörésének szélessége a csempe méretéhez képest.

**Bevel Slope Adjust** - Az élettörés meredeksége, lejtése.

**Grout Width** - A csempék közötti kötőanyag szélessége, szintén a csempe méretéhez viszonyítva.

**Grout Color** - A kötőanyag színe. Ha valamely színösszetevő negatív, nem jön létre külön szín.

**Bevel Smooth** - Az élettörés simasága, lágysága. Az 1.0-es érték hatására lesz a leglágább a csempe éle.

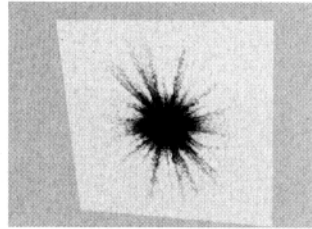
**Stagger Adjust** - Minden második sort (az X tengely mentén) eltolhatunk, ezáltal a csempék kötésbe hozhatók. Az eltolás mértéke a csempe szélességéhez van viszonyítva. A 0.5 hatására pont fél szélességgel tolódnak el az egyes csempesorok.

**Apply as Cheks** - Bármilyen pozitív érték hatására a textúra csak minden második csempén lesz hatással, ezáltal sakktáblaszerűen fognak elhelyezkedni a lesarkított és a sarkítatlan csempék.

### BLAST

Típus: Color, Reflect, Filter

A textúra égő, robbanásban lévő tárgy hatását kelti, mint például a puska torkolattüze. Hatása a tengelyközéppont körül a textúra tengelyének befoglaló keretén belül jelentkezik. Három színt adhatunk meg, ebből egy a robbanás középpontjában érvényesül, kettő pedig az innen kiinduló sugarakban.



Paraméterek:

**Radial Lines 1, 2** - A középpontból kitörő sugarak színei, amelyek keverednek is egymással.

**Central Color** - A robbanás középpontjának domináns színe.

**Radial Lines** - A robbanás középpontjából kiinduló sugarak száma. Értékének páros számnak kell lenni.

**Radial Line Size Vary** - Az egyes sugarak hosszának változatossága. Ha értéke 0.0 minden sugár azonos hosszú lesz.

## Textúrák

---

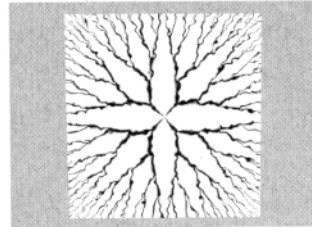
**Filter Adjust** - A robbanás területének filter módosítása.

**Noise** - A robbanás alakjának véletlenszerűsége.

### BMPBRNCH

Típus: Color, Reflect, Filter, Bump

Ez a textúra lágy élű, domború mintázatot rajzol, ami hasonló a növények ágainak burjánzásához. Az ágak a textúra közepéből indulva ágaznak szét. A textúra hasonló működésű, mint a Branches, vagy a DancSprk. Az ágak vastagsága szabályozható, két noise paraméterrel pedig a szabályosságuk csökkenthető. A szín, reflect és filter attribútum beállítható az ágakra. A mintázat mérete és elhelyezkedése a textúra tengelyével szabályozható.



Paraméterek:

**Branches** - A textúra középpontjából kiinduló ágak száma.

**Branch Width (0..1)** - Az ágak szélessége a textúra által lefedett területhez képest. Ha értéke 1.0, az ágak teljesen egybe érnek, nem lesz közöttük szabad hely.

**Branch Color/Reflect/Filter** - Az ágak attribútumai.

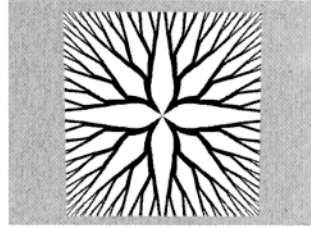
**Noise 1, 2** - Az ágak által alkotott mintázat változottságát kialakító noise paraméterei.

**Bump Adjust** - Az ágak magassága, vagy mélysége.

## BRANCHES

Típus: Color, Reflect, Filter

A textúra majdnem teljesen azonos a BumpBrnch-val, a különbség mindössze annyi, hogy az ágak nem lágyan, hanem éles kontúrral domborodnak ki. Jól használható háttérben álló fák koronájának imitálására, ha teljesen átlátszó korongra alkalmazzuk. Az ágak vastagsága és szabálytalansága paraméterezhető, Color, Reflect és Filter attribútumuk beállítható. A mintázat mérete, elhelyezkedése a textúra tengelyével szabályozható.



Paraméterek:

**Branch Width (0..1)** - Az ágak vastagsága a textúra által lefedett területhez képest.

**Branch Color/Reflect/Filter** - Az ágak attribútumai.

**Noise 1, 2** - Az ágak változatosságát kialakító fraktál zaj paraméterei.

**Branches (2,4,8..)** - A textúra középpontjából kiinduló ágak száma. Bármilyen érték használható, de páros szám esetén az ágak négyzetes területet futnak be.

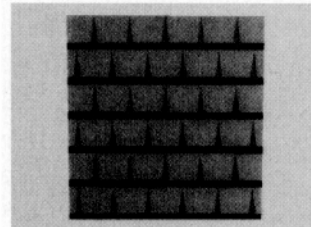
## SHINGLES

Típus: Bump

A textúra bump mintázatot alakít ki a tárgyon, amely a cseréptetőre hasonlít. A cserепek az X-Y síkon helyezkednek el. A cserép csúcsa +Y irányba mutat.

Paraméterek:

**X, Y, Z Size** - A textúra által létrehozott cserепek mérete.



## Textúrák

---

**Shingle Ht/Width** - A cserepek magassága és alsó élük szélessége a cserép teljes méretéhez képest.

**Face Slope Adjust** - Az egyes cserepek ferdesége, esése.

**Bottom/Left/Right Edge Slope** - A cserepek élettörése a három látható oldalukon.

**Stagger Adjust** - Az egyes cserépsorok eltolása X mentén, a csempe szélességéhez képest.

**Slope Variance** - Az élettört felületek variációja, változatossága.

**Shingle Size Vary** - Az egyes cserepek méretének állandósága, vagy változatossága. 0.0 vagy negatív érték esetén a cserepek azonos méretűek lesznek.

**Apply as Cheks** - Bármilyen pozitív érték hatására a textúra csak minden második cserépen érvényesül.

### STAMPED

Típus: Color, Reflect, Filter, Bump

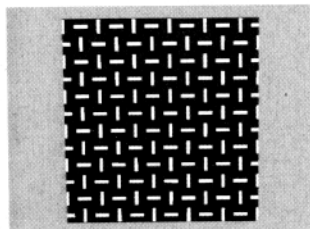
A textúra a BathTile leszárma-zottja, bordázott felületet hoz létre, amely hasonló a betonvas, vagy az acél padlólap mintázatához. Az egyes blokkok egymásra merőlegesen fekszenek. A mintázat az X-Y síkon helyezkedik el és mindkét irányban függetlenül méretezhető, a bordázat attribútumai megadhatók.

Paraméterek:

**X, Y Size** - A mintázat mérete a két tengely mentén egymástól függetlenül.

**Bevel Width** - A kiemelkedő mintázat élettörésének mértéke egy minta méretéhez viszonyítva.

**Bevel Slope Adjust** - Az egyes kiemelkedő (vagy be-süppedő) blokkok élettörésének meredeksége.



**Grout Width** - A mintázat kiemelkedései között lévő módosulatlan terület szélessége.

**Grout Color** - A mintázat blokkjai közötti terület színe.

**Bevel Smooth (0..1)** - Az élettörés lágysága.

**Grout Reflect/Filter** - A blokkok közötti rész eredeti Reflect és Filter értékeinek módosítása.

**Stagger Adjust** - A mintázat sorainak egymáshoz képest, az X mentén történő eltolása egy blokk méretének arányában.

**Apply as Checks** - Bármilyen pozitív érték hatására a textúra csak minden második blokkon fejt ki hatását, ezáltal csak az egy irányban elhelyezkedő blokkok jelennek meg.

### TRICHEKS

Típus: Color, Reflect, Filter, Bump

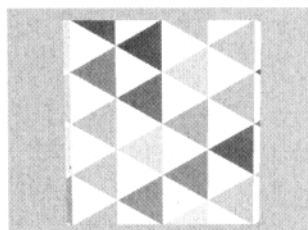
Ez egy másik "kockás" textúra, de itt az egyes elemek nem négyzetek, hanem háromszögek.

A méretük X és Y irányban egymástól függetlenül állítható, Z mentén nincs hatása. A lapok színezéséhez egy bázis színt, vagy az eredeti színt használhatjuk. Az aktuális szín ennek a variálásával alakul ki. Minden második lap színe a tárgy eredeti színével azonos, a többi a variációnak megfelelő véletlen színeket vesz fel. Az összes elem Reflect és Filter attribútuma véletlenszerűen variálható, csakúgy, mint az egyes elemek orientációja.

Paraméterek:

**X, Y Size** - Az elemek mérete.

**Color** - A textúra bázisszíne.



## Textúrák

---

**Red/Green/Blue Vary** - A bázis szín színösszetevőinek variációja, amely kialakítja az elemek aktuális színét. Nulla, vagy negatív érték esetén nem variálódik az adott összetevő.

**Color 'Value' Vary** - A textúra által kialakított szín variációjának a mértéke. Ezzel a paraméterrel nem színösszetevőnként, hanem egyben tudjuk variálni a textúra színét. Használhatjuk mindkét varianciát egyszerre is. Nulla, vagy negatív érték hatására minden elem azonos színű lesz.

**Reflect/Filter Level Vary** - Az elemek Reflect és Filter értékeinek maximális módosítása. Az eredeti paraméterek legfeljebb ekkora mértékben, véletlenszerűen módosított attribútumokat kapnak.

**Plate Angle Vary** - Az elemek felületének orientációja véletlenszerűen módosítható ennek a paraméternek a segítségével. Ha értéke 0.0 vagy negatív, minden felületelem azonos irányba néz.

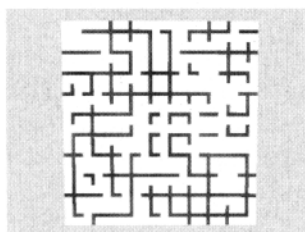
**Apply as Checks** - Bármilyen pozitív érték hatására a textúra csak minden második elemen fejt ki hatását, kínai sakktabla-szerű hatást kialakítva.

### TRACER

Típus: Color, Reflect, Filter, Bump

Ez a textúra függőleges és vízszintes vonaldarabokból techno látványú mintázatot alakít ki.

A vonaldarabok attribútuma interpolálódik a textúra tengelyközépponttól való távolságának értékével, hasonlóan, mint a ColrNoiz és a BumNoiz textúráknál. A textúra mérete és elhelyezkedése a tengelyén keresztül szabályozható. A vonalszegmensek attribútumai beállíthatók, de származhatnak az eredeti értékekből is.



Paraméterek:

**Wall Width Adjust** - A vonalak által lefedett terület aránya a textúra által lefedett területhez képest. Magasabb érték vastagabb, több vonalat eredményez.

**Color 1, 2** - A mintázat bázisszíne. Ezt variálja a tengelyponttól való távolság arányában.

**Reflect/Filter 1, 2** - A vonaldarabok Reflect és Filter attribútumainak mértéke.

**Bump Adjust** - A vonalak esetleges egyenetlensége.

**Noise 1, 2** - A vonalak változatosságát kialakító fraktál zaj paraméterei.

### TRITILE

Típus: Color, Bump

Hasonló mintázat, mint a csempézett fal, de a csempék alakja háromszög.

Paraméterek:

**X, Y Size** - A csempék mérete, egymástól függetlenül.

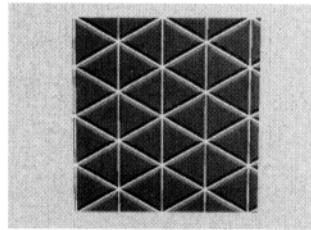
**Bevel Width** - A csempe élettörésének mérete a csempe méretéhez képest.

**Bevel Slope Adjust** - Az élettörés meredeksége.

**Grout Width** - A csempék közötti kötőanyag vastagsága a csempe méretéhez képest. Nulla, vagy negatív érték hatására nem keletkezik kötőanyag.

**Grout Color** - A kötőanyag színe. Ha valamely színösszetevő értéke negatív, a kötőanyag színe azonos lesz a tárgy eredeti színével.

**Apply as Checks** - Bármilyen pozitív érték hatására a textúra csak minden második csempén alkalmazódik.

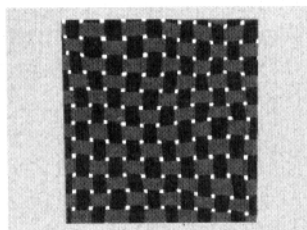




### WEAVE

Típus: Color, Reflect, Filter, Bump

Ez a textúra a szín és a bump attribútum módosításával hasonló felületet hoz létre, mint a nádfonatú székek bevonata. A vízszintes és függőleges szálak színe külön állítható, csakúgy, mint a szálak szélessége. A mintázat egyenetlensége széles határok között állítható, negatív érték is lehetséges, de ekkor a csillogó foltok rossz helyre kerülése miatt csalóka kép fog kialakulni.



Paraméterek:

**X, Y Size** - A textúra mérete a két tengely mentén. Z irányban nincs hatása.

**Stripe Width** - A szálak szélessége a mintázat X/Y méretéhez viszonyítva.

**Noise 1, 2** - A mintázat változatosságának paraméterei.

**Bump Adjust** - A szálak egyenetlenségének mértéke azokon a pontokon, ahol a szálak keresztezik egymást.

**Horiz./Vert Stripe Color** - A vízszintes és függőleges szálak színe. Negatív érték hatására az adott szálak a tárgy eredeti színét veszik fel.

**Stripe Reflect/Filter** - A szálak fényvisszaverő és fényáteresztő képességének mértéke. Negatív érték esetén ez azonos lesz a tárgy eredeti attribútumával.

### Alakfüggő textúrák

Ebbe a csoportba olyan textúrák tartoznak, amelyek speciálisan egy-egy meghatározott alakú tárgyon fejtik ki hatásukat. Felhasználhatók ugyan más alakokon is, de hatásuk nem lesz olyan látványos, vagy nem azt nyújtják, mint amit eredetileg kellene. Ezek a textúrák általában többféle primitív típushoz is elkészültek, például ugyan annak a mintázatnak lehet speciálisan síklapra, hengerre és gömbre írt változata.

Az egy családba tartozó textúráknak vannak általános paraméterei, amelyek azonosak minden tagnál, és vannak egyedi paraméterei, amelyek csak a család egy tagjára jellemzőek. A leírásban az általános paramétereket a család ismertetésénél összefoglalva, az egyedi paramétereket pedig a textúrák leírásánál ismertetem.

### CHEKS család

Típus: Color, Reflect, Filter, Bump

A textúra kockás mintázatot visz a tárgyra. A kockák függetlenül méretezhetők minden, az adott textúránál lehetséges irányban. A színezéshez egy bázisszín használható, amelyet színösszetevőnként, vagy egyben variálhatunk. A fényvisszaverő és fényáteresztő képesség szintén módosítható véletlen algoritmus szerint. Lehetőség van a kockák orientációjának variálására is.

Általános paraméterek:

**Color** - A textúra bázisszíne.

**Red/Green/Blue Vary** - Az egyes színösszetevők varianciája. Ekkora határon belül véletlen arányban változik a bázisszín az egyes kockákon.

## Textúrák

---

**Color 'Value' Vary** - A textúra bázisszínének varianciája. Ezzel nem színösszetevőnként, hanem egészében tudjuk variálni a bázisszínt.

**Reflect/Filter 'Level' Vary** - A Reflect és Filter attribútumok varianciája. Nulla, vagy negatív érték használatával az eredeti attribútumok nem változnak.

**Plate Angle Vary** - Az egyes kockák orientációja véletlen értékekkel módosítható. Ez a paraméter a módosítás maximuma két szomszédos kocka között.

**Apply as Checks** - Bármilyen pozitív érték hatására a textúra csak minden második kockán fejt ki hatását, ami sakktáblaszerű mintázatot eredményez.

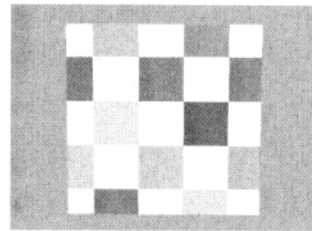
### RECTCHEKS

Ez a textúra kifejezetten négyzetes tárgyakon való alkalmazásra készült. A kockasorok Y mentén egymáshoz képest eltolhatók, téglaszerű hatást kialakítva.

Paraméterek:

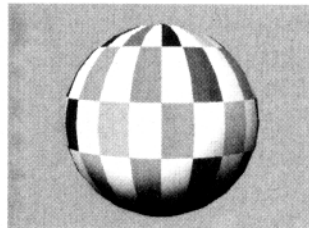
**X, Y, Z Size** - A textúra méretének paraméterei.

**Stagger Adjust** - Az egyes sorok eltolásának a mértéke.



### RADCHEKS

A textúra ezen változata a textúra középpontjából kiindulva radiálisan fejt ki hatását, ezért legjobb hatást forgástesteken, például gömbön, hengeren érhetünk el vele.



Használható síkon is, ekkor a mintázat a Z tengely körül koncentrikusan jön létre.

Paraméterek:

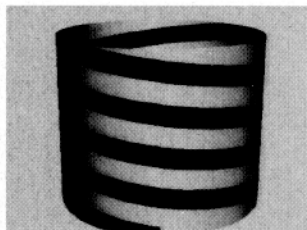
**Radial Scale** - A mintázat mérete sugárirányban.

**Z Scale** - A textúra mérete Z irányban.

**Sweep Division** - A textúra mérete Z körül, vagyis hogy hány kocka helyezkedjen el egy gyűrűn.

### SPRLCHEKS

Ez a textúra henger alakú tárgyakra spirális mintázatot rajzol, hasonló, mint ami a tengerparti világítótornyokon látható. A mintázat a Z tengely mint középpont körül alakul ki, ezért a legjobb hatást akkor érhetjük el, ha a Z tengely a tárgy közepére esik és hosszirányba mutat.



Paraméterek:

**Radial Scale** - A textúra mérete sugárirányban.

**Z Scale** - A mintázat mérete Z irányban.

**Sweep Division** - A textúra mérete Z körül, vagyis a spirálok száma.

### DEATHSTAR család

Típus: Color, Reflect, Filter, Bump

A textúra véletlen rendben elhelyezkedő lapokat helyez a tárgyra, amely olyan, mint egy gép külső, ütött-kopott lemezelése. Legjobb példa a (neve is erre utal) StarWars halálcsillaga. Életörés alkalmazható, erre külön színt is megadhatunk, ezen kívül az elemek sarkában egy szennyeződést imitáló színt is tehetünk. Az elemek bump mappingja kikapcsolható, ekkor a nyomtatott áramkörökéhez hasonló látványt érhetünk el.

## Textúrák

---

Általános paraméterek:

**X, Y, Z Eccentricity Adj** - Az elemek méretének változatossága a három tengely mentén külön-külön. A beállított méretek ennek a paraméternek a függvényében változnak az egyes elemeken.

**Bevel Width Adjust** - Az elemek élettörésének szélessége. Ez a terület helyezkedik el a tulajdonképeni csempék között.

**Bump Adjust** - Az elemek magassága. Ha értéke 0.0, akkor az elemek egy síkban maradnak, az élettörés pedig a köztes rész lesz, mint a csempéknél a kötőanyag.

**Red/Green/Blue Bevel Color** - Az élettörés színe. Ez a terület felfogható úgy is, mint az elemek közötti rész.

**Bevel Reflect/Filter** - Az élettörés Reflect és Filter mértéke. Negatív érték esetén az a paraméter azonos lesz a tárgy eredeti attribútumával.

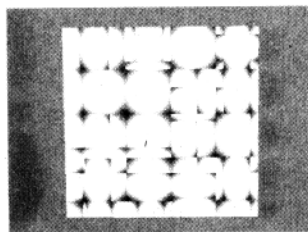
**Bevel Dirt/Rust** - Ezekkel a paraméterekkel piszkos és rozsdás foltokat tehetünk az elemek sarkába. A méretet az egész elem méretéhez viszonyítva kell megadni. Ha mindkét funkciót használod, a legjobb eredményt az adja, ha a rozsdá (Rust) mérete kisebb, mint a piszoké (Dirt).

### DETHSTAR (DeathStar)

A textúra ezen változata négyzetes tárgyakon való használatra lett tervezve.

Paraméterek:

**X, Y, Z Size** - A textúra méretei a tengelyek mentén, egymástól függetlenül.



## RDDTHSTR (Radial

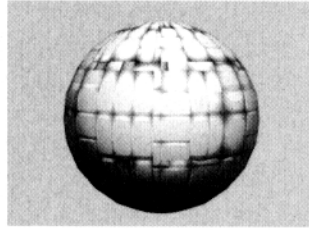
DeathStar)

Ez a változat radiális elhelyezés esetén nyújtja a legjobb eredményt, ezért hengeren, vagy gömbön alkalmazzuk. A mintázat a Z tengely körül alakul ki.

**Z Scale Adj** - A csempék mérete a Z tengely mentén.

Radial Scale Adjust - Az elemek mérete a textúra sugara mentén.

**Sweep Divs** - A csempék mérete a Z tengely körül, vagyis a kerületen elhelyezkedő elemek száma. Értelmen szerint ez egész szám esetén adja a legjobb eredményt, de ez nem előírás, tört számot is megadhatunk.



## HONEY COMB család

Típus: Color, Reflect, Filter, Bump

A textúra család tagjai méhsejtszerűen elhelyezkedő hatszögekből építik fel a mintázatot. A HoneyComb és a RadComb méhsejtek színéül a tárgy eredeti színét használják, míg a Hexez ezt a hatszögek közepén alkalmazza. A másik szint a textúra a kérdezőjében megadott szín variálásával képzí. Az alkotó elemek méretét és elhelyezkedését a textúra tengelyével tudjuk befolyásolni. Egy bump paraméter segítségével a méhsejtek egyenetlenné is tehetők.

Általános paraméterek:

**Wall Width Adj (0..1)** - A cellák falának vastagsága a cella méretéhez viszonyítva. Ha 0.0-s értéket adunk meg, nem lesz falvastagsága a celláknak, tehát a méhsejt sem jön létre.

## Textúrák

---

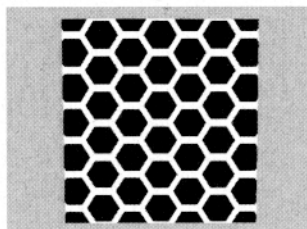
**Color 1, 2 R/G/B** - A méhsejtek közötti rész (Hexez esetén a falának) a bázisszínei, a valós szín e két érték véletlen variációjából alakul ki.

**Reflect/Filter 1, 2** - A méhsejtek közötti rész (Hexez esetén a falának) Reflect és Filter mértéke. Az aktuális érték e két paraméter variálásával jön létre. Negatív értékkel az adott paraméter azonos lesz a tárgy eredeti attribútumával.

**Bump Adjust** - A cellák egyenetlenségének mértéke. Negatív érték, vagy 0.0 alkalmazásával a cellák éle egy síkban lesz.

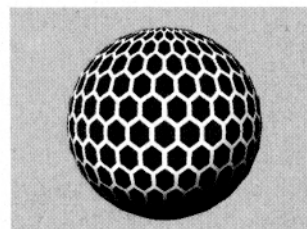
### HONYCOMB

Ez a változat négyzetes tárgyon, vagy síklapon való alkalmazáshoz készült, lépes méz kialakításához ideális. A mintázat az X-Y síkon jön létre, a méhsejtek mélysége Z-vel párhuzamos.



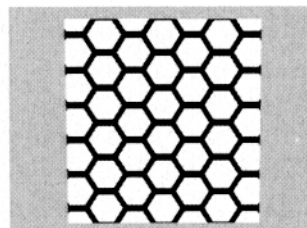
### RADCOMB

A textúra hengeres, vagy gömbszerű tárgyakon használható eredményesen. A mintázat a Z tengely körül alakul ki.



### HEXEZ

A HoneyComb-hoz hasonlóan négyzetes tárgyakon alkalmazható, a különbség, hogy nem a "méz", hanem a cellák falának színe variálódik, a cellákban lévő anyag attribútuma fog a tárgy eredeti attribútumából származni.



### PLAID

Típus: Color

A textúra azt teszi, mint amit a neve is sejtet, plédet hoz létre. Ehhez három színt, valamint zaj és eltolási paramétereket használ fel. Ez nem család, hanem önálló textúra, amely legjobban négyzetes tárgyakon érvényesül.

Paraméterek:

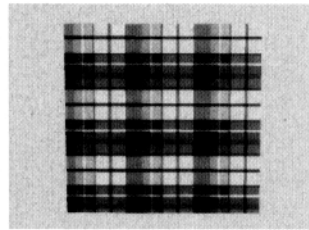
**X, Y, Z Size** - A textúra mérete a három tengely mentén önállóan.

**Color 1, 2, 3** A mintázat kialakításához használt három szín. Valamely szín bármely összetevőjének negatív értéket adva, az adott színt a tárgy eredeti attribútuma határozza meg.

**Noise** - A mintázatot kialakító fraktál zaj paraméterei.

**Stagger Adjust** - Az egyes mintasorok eltolása a textúra X tengelye mentén.

**Apply as Cheks** - Bármilyen pozitív érték hatására a textúra csak minden második kockán fejt ki hatását, sakktáblaszerű kinézetet adva a mintának.



### WINDOW család

Típus: Color, Reflect, Filter

A családba három textúra tartozik. Ezek a tárgy felületére véletlen mintázat szerint elhelyezett ablakokat hoznak létre, amiktől a tárgy hasonlóan néz ki, mint egy felhőkarcoló, vagy egy űrhajó. Az ablakokat a tárgy a felület átlátszóságának változtatásával képezi.



## Textúrák

---

Általános Paraméterek:

**Low/Hi Treshold** - Az ablakok által lefedett terület aránya a teljes felülethez képest. A lefedett területet a két paraméter különbsége határozza meg. Ha a teljes felületet 100%-nak vesszük, a 0-Lo és Hi-1 közötti felületrész-arányok az ablakok közötti területek felületrésze, míg a Lo-Hi részek lesznek az ablakok.

**Min/Max Spacing** - Az ablakok közötti minimális és maximális távolság, ha a két érték azonos, minden ablak egyforma messze lesz egymástól.

**Transparency** - Az ablakok átlátszóságának mértéke. Alap esetben az ablakok a felület átlátszóságának módosításával jönnek létre de lehetőség van a színük változtatására is. Ha az átlátszóság értéke negatív, akkor csak a színe változik az ablakoknak.

**Color 1, 2** - Az ablakok két bázisszíne. Az egyes ablakok színe e két érték interpolálásával véletlenszerűen alakul ki. Ha bármelyik közülük negatív, akkor az helyett a tárgy eredeti színe kerül alkalmazásra.

**Reflect Adjustment** - Az ablakok által létrehozott Reflect variációjának a mértéke. Ha negatív értéket használunk, akkor a tárgy eredeti fényvisszaverő képessége nem változik.

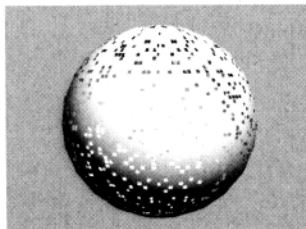
### RADWINDS

A textúra ezen változata speciálisan gömbszerű tárgya-kon való alkalmazáshoz készült.

Paraméterek:

**Radial Scaling** - Az ablakok mérete a textúra sugarának irányában.

**Z Scaling** - Az ablakok mérete Z irányában.



**Sweep Division Adjust** - Az ablakok mérete a kerület mentén, vagyis a kerületen elhelyezkedő ablakok száma. Egész szám adja a legjobb eredményt.

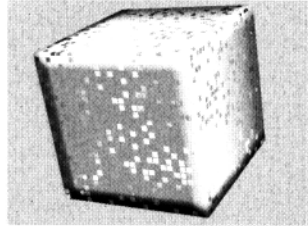
### RECTWIND (Rectangle

Windows)

Ez a textúra négyzetes tárgyakon való alkalmazásra készült.

Paraméterek:

**X, Y, Z Size** - Az ablakok mérete a három tengely entén külön-külön.



### TUBEWIND

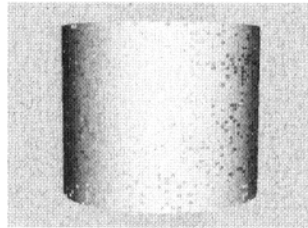
Ez a változat hengeres tárgyakon való alkalmazásra készült, legjobb eredmény a Tube primitíven adja. Az ablakok a Z tengely körül körkörösön jönnek létre.

Paraméterek:

**Radial Scaling** - Az ablakok mérete sugár irányban.

**Z Scaling** - Az ablakok magassága Z mentén.

**Sweep Division Adjust** - Az ablakok száma a kerület mentén. A legjobb eredményt egész szám alkalmazásával érhetjük el, de tört szám is megadható.



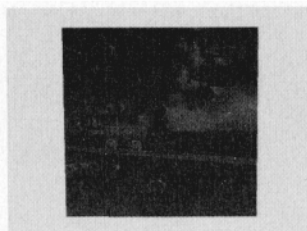
## Speciális alkalmazású textúrák

Ebbe a csoportba olyan textúrák tartoznak, amelyek alkalmazása eltér a szokványostól. Van olyan textúra például, amelyik a tárgyra feszített brush-t módosítja, úgy, hogy az antik, megsárgult kinézetű lesz, vagy van olyan, amely viaszos csillogásúvá teszi a tárgyat, olyanná, mint egy friss alma.

### ANTIQUE

Típus: Color, Reflect, Filter

Ez a textúra bump brus-al együtt használható, segítségével az egyenetlenség aljára szennyeződéshez hasonló színezetet tehetünk, amely régies, antik kinézetet ad a tárgynak. Használatához ugyan azt a képet, azonos elhelyezkedésben fel kell vinni a tárgyra bump és color mapként, majd utánuk alkalmazni ezt a textúrát. A bump map a szokásos módon fejt ki hatását, de a color map helyett a textúra kérdezőjében beállított attribútumok fognak érvényesülni, a color map vörös összetevője pedig a textúra alapjául fog szolgálni. Ennek segítségével határozza meg a textúra, hogy hol vannak a kiemelkedések közötti mélypontok, ahol zajjal kombinálva a színeket a fekete felé módosítja, kialakítva ezzel a szennyezett látványt. Leginkább fényes anyagú, pl. réz, króm tárgyakon, ray-trace módban kiszámoltatott képeken érvényesül.



Paraméterek:

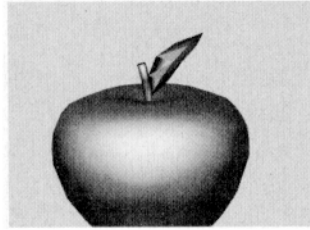
**Color/Filter/Reflect** - A tárgy azon részének attribútumai, amelyet a textúra lefed. Ez nem azonos az antik hatás attribútumaival, az úgy jön létre, hogy a megfelelő helyeken a színeket a matt fekete felé módosítja.

**Noise Magnitude, Velocity** - A szennyezett hatást módosíthatjuk egy fraktál zajjal, ezek ennek a paraméterei. Ha minkét input mezőbe 0.0-t írunk be, nem fog érvényesülni a zaj.

### CNDYAPPL (Candy Apple)

Típus: Color

Ez viaszos kinézetet ad a tárgynak, hasonló hatást létrehozva, mint ami a frissen szedett, hamvas gyümölcsökön, például az almán látható. A textúrához nem tartozik paraméter, teljes egészében a beépített algoritmus alapján dolgozik.



### COOLFIR

Típus: Color, Reflect, Filter

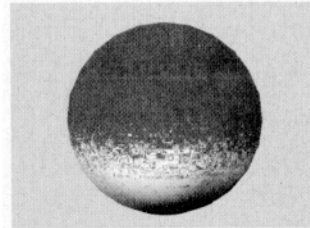
Ez a textúra hasonló a régebbi

Linear textúrához, A tárgy -Z irányban fekvő eredeti színét úsztatja át +Z irányában a para-

méterek között megadott színbe. A különbség, hogy ez méretezhető és mozgatható a textúra tengelye által, valamint a színátmenet befolyásolható egy Noise paraméterpárossal, így nem csak sima színátmeneteket nyerhetünk.

Paraméterek:

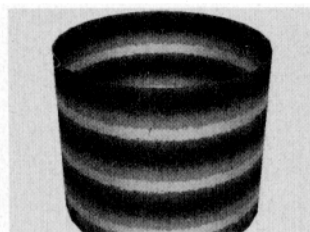
**Color/Reflect/Filter** - A textúra +Z irányába eső attribútumai, ezekbe úsznak át -Z irányból az eredeti attribútumok. Bármely paraméter valamely színösszetevőjének negatív értéket megadva, az adott attribútum nem vesz részt az átalakításban, a tárgy egészen az eredeti értékével szerepel.



### CRAKS

Típus: Color, Reflect, Filter, Bump

Ez a textúra azzal a céllal íródott, hogy megkönnyítse az egyforma blokkokból felépített pillérek létrehozását. A textúra ismétlődő sávokban bump mintázatot visz a tárgyra, amelyet fraktál zaj segítségével variálni is tud. A mintázat ismétlődését a textúra Z tengelyének méretével tudjuk szabályozni. A mintázat által létrehozott, a blokkokat elválasztó sávok szélessége a kérdezőben megadott értékkel szabályozható.



Paraméterek:

**Fract Colored (0..1)** - A blokkok közötti elválasztó sáv szélessége a blokk méretéhez képest. Ha értéke 0.0, nem lesz elválasztó sáv, ha 1.0, akkor nem lesz blokk, csak elválasztó sáv.

**Noise 1, 2** - Az elválasztó sávok széleit egyenetlenné tehetjük e két fraktál noise paraméterpárossal. Ha minden noise érték 0.0, akkor az elválasztó sávok teljesen szabályos szélűek lesznek.

**Filter/Reflect Adjust** - Az elválasztó sávok fényvisszaverő és fényáteresztő képességének mértéke. Negatív érték esetén azonos lesz az eredeti attribútummal.

**Crack Color** - A sávok színe.

**Bump Adjust** - A sávok magassága/mélysége. Ez az érték módosítva lesz az esetleges noise értékkel.

**Bump Smoothing (0..1)** - Az elválasztó sávokon kialakuló bump szélének lágytsága.

**Colr Intensity (0..1)** - Ez a paraméter azt mutatja meg, hogy a tárgy eredeti attribútuma milyen erősen látszik át a textúra által felvitt attribútumon.

**Color Smoothing (0..1)** - Ezzel a paraméterrel azt tudjuk szabályozni, hogy a textúra által felvitt színek milyen éles kontúrral illeszkedjenek a tárgyon lévő színhez. Ha értéke 1.0, akkor lesz a leglágyabb a színátmenet.

### DASHLINE

Típus: Color, Reflect, Filter

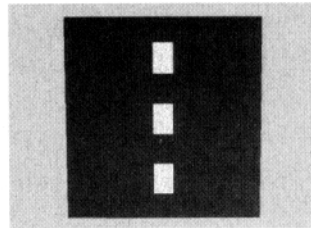
Ezzel szaggatott vonal mintázatot hozhatunk létre a tárgyon, mint ami az utakon látható. A vonalak Y mentén helyezkednek el, méretük beállítására a textúra tengelye szolgál. A textúrához 3 noise paraméter tartozik, kettő a szokásos noise, amely a mintázat alakját befolyásolja, egy pedig a Color Vary, amely a szín változottságáért felelős. Utóbbival olyan hatást hozhatunk létre, mintha a terelővonal kissé lekopott volna az útról. Arra is van mód, hogy meghatározzuk a terelővonal felfestéséhez használt festék intenzitását. Ezt alacsonyabbra véve a tárgy eredeti színe megfelelő mértékben átlátszik a mintázaton is.

Paraméterek:

**Fract Colored (0..1)** - Értéke a vonalszakasz hosszát határozza meg. A mintázat egy ismétlődő eleme egy vonalszakaszból és egy, a két vonal közé eső üres területből áll. Ezek együttes méretét a textúra Y tengelyének hossza határozza meg. A Fract Colored annak az aránya, amennyi ebből a méretből a vonalszakaszra eső hossz.

**Shape Noise 1, 2** - A vonalak szélének egyenetlenségét alakító zaj paraméterei.

**Filter/Reflect Adjust** - A vonalszakaszok fényvisszaverő és fényáteresztő tulajdonságának a mértéke.



## Textúrák

---

**Dash Color** - A szaggatott vonal színe.

**Color Vary, Color Vary Vel (0..1)** - A vonal színének változatosságát kialakító fraktál zaj paraméterei.

**Dash Intensity (0..1)** - A vonal színének erőssége. Alacsonyabb érték használata esetén a vonal festéke kevésbé fed le a tárgyat, annak színe átlátszik a vonalon.

### FIREBALL

Típus: Color, Reflect, Filter

A textúrával lángoló tárgy hatását érhetjük el. A láng két színből fog kialakulni, egyik szín a tűz középpontjában domináns, a másik a láng szélén. A tűz a textúra tengelypontjából gömbszerűen terjed szét sugár irányban, ezért olyan gömbön, amelynek a tengelypontja épp a közepén van, nincs látható hatása, Ellenben ha a tengelypontot elmozgatjuk például a gömb aljára, már látható a tűz. Fokozza a hatást, ha a gömböt egy irányban megnyújtjuk, és a textúra tengelyei is olyan hosszúak, hogy minden irányban a tárgy széléig érnek. A tűz színe a textúra tengelyeinek távolságain alakul át teljesen.

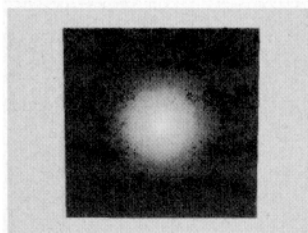
Paraméterek:

**Color 1** - A tűz közepének a színe

**Color 2** - A tűz szélének a színe.

**Reflect/Filter 1, 2** - A tűz közepének és szélének attribútumai. A tulajdonságoknak csak az intenzitása adható meg, a színük nem, ez a Color értékekből képződik.

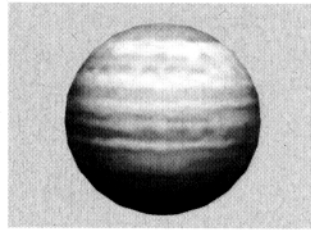
**Noise** - A két szín közötti színátmenetet módosító zaj paraméterei.



## GASGIANT

Típus: Color, Filter

A textúra Z körül koncentrikusan elhelyezkedő gyűrűket helyez a tárgyra, amelyek a tárgyat szétszabdalni látszanak. A gyűrűk szélének szabálytalanságát noise paraméterrel befolyásolhatjuk. Megfelelően kiválasztott színek esetén jól utánozható a textúrával a Szaturnusz bolygó. A bolygó gyűrűit egy korongra helyezett GasGiant textúrával hozhatjuk létre.



Paraméterek:

**Color 1, 2** - A gyűrűk két bázisszíne. Ezek véletlen interpolációjával alakulnak ki a valódi színek.

**Banding Width Adjust** - A gyűrűk átlagos szélessége.

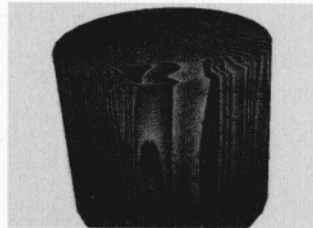
**Filter Adjust** - A gyűrűk átlátszóságának mértéke. Azt mutatja meg, hogy a gyűrűk átlátszósága milyen messze esik a teljes átlátszóságtól. Minél nagyobb tehát az értéke, annál kevésbé ereszti át a fényt a gyűrű. Ez a paraméter nem vonatkozik a gyűrűk közötti hézagokra, azok átlátszóságát a tárgy eredeti attribútuma határozza meg.

**Noise** - A rostok szélének egyenetlensége.

## HARDWOOD

Típus: Color, Bump

A textúrával keményfa mintázatot tudunk generálni. A fa elsődleges színét a tárgy attribútuma határozza meg, az erezet színe pedig a textúra kérdezőjében adható meg. A gyűrűk a Z tengely körül alakulnak ki, egyenetlenségük noise paraméterrel szabályozható.





## Textúrák

---

Paraméterek:

**X, Y, Z Size** - A fa évgyűrűinek mérete.

**Noise 1, 2** - A rostok alakjának változatossága, egyenetlensége.

**Bump Adjust** - A fa mintázatának térbeli egyenetlensége azokon a helyeken, ahol felszínre kerül.

**2nd Wood Color** - A fa másodlagos színe.

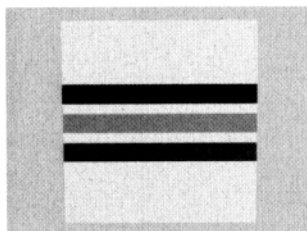
**Grain Color** - A fa rostjainak a színe.

**Grain Spacing Adj** - A párhuzamos rostszálak közötti távolság átlagos mérete.

### HRDSTRIP (Hard Stripes)

Típus: Color, Reflect, Filter

Ez a textúra párhuzamos csíkokat rajzol a tárgyra, maximum hármat. A csíkok az X-Y síkon jelennek meg, az X tengellyel párhuzamosan. tehát sík lapon alkalmazva Z-nek a tárgyból kifelé kel mutatni.



Paraméterek:

**Stripe 1, 2,3 Width** - A csíkok szélessége.

**Stripe 1, 2 Space** - A csíkok után következő szabad hely szélessége.

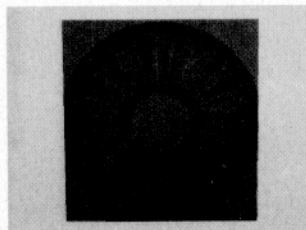
**Stripe 1, 2, 3 R/G/B** - A három csík színe. Bármelyik színösszetevőnek negatív értéket adva az a csík nem jelenik meg.

**Stripe Reflect/Filter** - A csíkok Reflect és Filter értéke. Ezeket nem lehet a három csíkra külön-külön megadni.

### IRIS

Típus: Color

Ez a textúra a szemgolyó íriszének utánzásához készült, mint ahogy a neve is mutatja. Legjobb



hatás olyan korongon alkalmazva érhetünk el vele, amelynek Z tengelye merőlegesen áll a felületre. A textúrához négy szint határozhatunk meg, ebből kettő bázisszín, amelyek interpolálásával kapjuk meg a pupilla és az írisz határvonala közé eső terület színét, a harmadik szín a pupillát körülvevő sáv színe, míg a negyedik az írisz külső vonalának színe. Egy noise paraméterpáros segítségével utóbbi két színsáv változatosságát szabályozhatjuk. A pupilla színe a tárgy színével azonos.

Paraméterek:

**Inner/Outer Radius** - A textúra méretei. Az Inner Radius a pupilla sugara, az Outer Radius pedig az írisz külső mérete.

**Iris Color 1, 2** - Az írisz két bázisszíne, ezek véletlen interpolálásával jön létre a valódi szín. Negatív értékű színösszetevő alkalmazása esetén az adott szín azonos lesz a tárgyon már kialakult színnel.

**Inner Color** - A pupilla és az írisz közötti sáv színe.

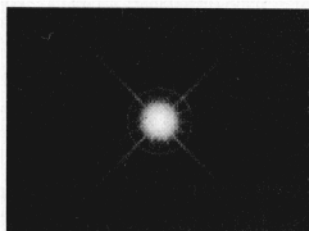
**Outer Color** - Az írisz külső szélének színe.

**Noise** - Az írisz két bázisszínének keveredését szabályzó véletlen algoritmus paraméterei.

### LENSFLAR (LensFlare)

Típus: Filter

A textúra a fényforrásba néző kamerában kialakuló csillanás modellezésére lett készítve. Olyan felületeken érdemes használni, amelyekből a Z tengely kifelé mutat. A textúra a felület átlátszóságát módosítja, így hozva létre a becsillanó hatást. A hatást négy különböző alaknak a tárgyra vitelével hozza létre, egy csillogó fátyol, egy gyűrű és egy négy, valamint egy nyolc ágú csillag jön létre a tárgyon. Mind a négy alak mérete és



## Textúrák

---

intenzitása szabályozható. A tárgy meglévő színe szolgál a csillagok színéül, még akkor is, ha az előtte egy, vagy több textúrával vagy brush-al módosítottuk.

Paraméterek:

**Flare Radius** - A belső csillogás sugara.

**Flare Inten (0..1)** - A csillogás intenzitása, vagyis hogy mennyire átlátszatlan. Minél nagyobb ez az érték, annál kevésbé átlátszó a csillogó rész.

**Ring Radius Adj** - A csillogás körül kialakuló gyűrű sugara.

**Ring Select (1, 2, 3 ...)** - A gyűrű mérete.

**Ring Inten (0..1)** - A gyűrű intenzitása.

**Star 1 Max Radius** - A nyolcágú csillag mérete.

**Star 1 Inten (0..1)** - A nyolcágú csillag intenzitása.

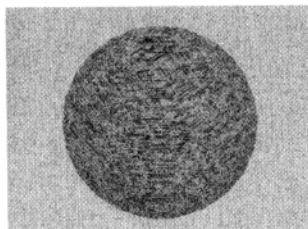
**Star 2 Max Radius** - A négyágú csillag mérete.

**Star 2 Inten (0..1)** - A négyágú csillag intenzitása.

## METALS

Típus: Color, Reflect, Bump

A textúra segítségével hat előre definiált fémattribútumot állíthatunk ba a tárgyra, hatféle felületmegtartási fokon. A felület fényességét és antik kinézetét szintén szabályozhatjuk.



Paraméterek:

**Metal Number** - A kívánt fém típusát választhatjuk ki 1-6 értékek valamelyikével. A kiválasztható fémek felsorolása a kérdezőben is megtalálható. Ezek: 1-arany, 2-sárgaréz, 3-vörösréz, 4-króm, 5-ezüst, 6-Alumínium.

**Dull/Polished** - A fém fényessége. Minél magasabb ez az érték, annál nagyobb lesz a fém fényvisszaverő képessége.

**Finish** - A fém megmunkáltságának hat fokozata van, ezek közül választhatunk ebben az input mezőkben. Az egyes értékekhez tartozó felületminőség szintén le van írva a kérdezőben. A lehetséges értékek: 1-sima, 2-egyenetlen, 3-göröngyös, 4-karistolt, 5-kavicsos, 6-horpadt.

**Antique** - Ez a paraméter antik hatást visz az egyenetlen fémfelületre. Ez abból áll, hogy az egyenetlenségek alján a tárgy színét sötétebbre veszi, szennyezett, régies hatást kiváltva ezzel.

### MNTNTOP (Mountain Tops)

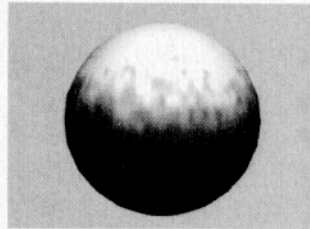
Típus: Color, Reflect, Filter

Ez a textúra hasonló a régi Linear-hoz, két attribútum között átmenetet hoz létre. Az egyik attribútum a tárgy eredet attribútuma, ez -Z irányban helyezkedik el, a másik a textúra kérdezőjében adható meg, ez lesz a +Z irányban. Lényeges különbség a két textúra között, hogy a MntnTop színátmenete noise paraméterrel variálható, valamint, hogy a színátmenet szélessége és elhelyezkedése a textúra tengelyével szabályozható. Segítségével hasonló hatást hozhatunk létre, mint a hegyek tetején lévő hósapka.

Paraméterek:

**Color/Reflect/Filter** - A textúra +Z tengelyének irányába eső attribútumok.

**Noise** - A színátmenet egyenletességét módosíthatjuk fraktál zaj segítségével, ezek ennek a paraméterei.



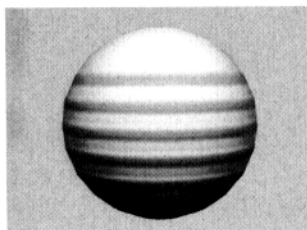
## Textúrák

---

### RIBBED

Típus: Color, Reflect, Filter, Bump

A textúra megadott számú bordát helyez a tárgy felszínére. A bordák színe, fényvisszaverő és fényáteresztő képessége megadható és lehetőség van a bumpolásukra is. A textúra csak a tengelynek befoglaló keretén belül jeleníti meg a bordákat, amelyek szélének lágysága, egyenetlensége és intenzitása szabályozható.



Paraméterek:

**Fract Colored (0..1)** - Az ismétlődő mintázaton belül a bordák aránya. Ha értéke 1.0, a bordák összeérnek, ha 0.0, nem lesznek bordák. A 0.5 érték hatására a bordák és a köztük lévő terület szélessége azonos lesz.

**Noise 1, 2** - A borda alakjának változatossága.

**Filter/Reflect Adjust** - A bordák fényvisszaverő és fényáteresztő képességének mértéke.

**Rib Color** - A bordák színe.

**Bump Adjust** - A bordák kiemelkedésének, vagy be-süppedésének mértéke.

**Bump Smoothing** - A bumpolt bordák élének lágysága, vagy keménysége. Az 1.0 érték adja a leglágyabban lefutó bordaélt.

**Colr Intensity (0..1)** - A bordák színének intenzitása. Alacsonyabb érték esetén a bordák alatt lévő, esetleg más textúrák és brush-ok által már módosított szín az intenzitástól függő mértékben átlát-szik.

**Number of Ribs** - A textúra tengelyeinek befoglaló keretén belül megjelenő bordák száma.

## SFTSTRIP (Soft Strip)

Típus: Color, Reflect, Filter

Ez a textúra lágy élű, párhuzamos csíkokat rajzol a tárgyra, olyat, mintha azok festékszóróval lennének felfestve. Egy textúrával maximum három csíkot rajzolhatunk fel. A csíkok az X-Y síkon jelennek meg, az X tengellyel párhuzamosan, tehát sík lapon alkalmazva Z-nek a tárgyból kifelé kel mutatni.

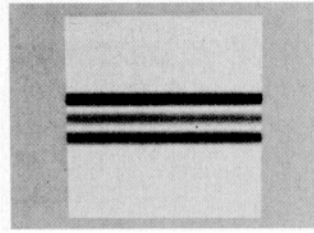
Paraméterek:

**Stripe 1, 2,3 Width** - A csíkok szélessége.

**Stripe 1, 2 Space** - A csíkok után következő szabad hely szélessége.

**Stripe 1, 2, 3 R/G/B** - A három csík színe. Bármelyik színösszetevőnek negatív értéket adva az a csík nem jelenik meg.

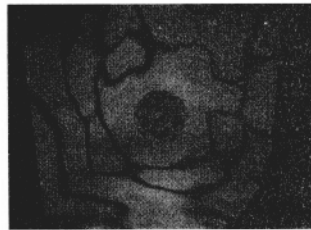
**Stripe Reflect/Filter** - A csíkok Reflect és Filter értéke. Ezeket nem lehet a három csíkra külön-külön megadni.



## TRANSPAR (Transparency)

Típus: Filter

A textúra csak egyetlen, igen egyszerű dolgot tesz, a tárgyon lévő színeket átalakítja filter értékekké, függetlenül attól, hogy az illető szín a tárgy alap attribútuma, esetleg textúra vagy brush hozta létre. Az így kialakított tárgyon áthaladó, árnyékvetés tulajdonsággal (Cast Shadow) rendelkező fény színe interferál a tárgyével, mint ahogy a valóságban a színes üvegen áthaladó fény teszi. Csak trace képszámítási módban érvényesül a hatása, mivel csak ekkor figyelni valóban a



## Textúrák

---

fénysugarak útját és az azokat ért módosító hatásokat a program.

Paraméterek:

**Transparency** - Az átlátszóság mértéke. A túl magas érték azt eredményezheti, hogy a tárgy egyáltalán nem lesz látható!

### VENITIAN (Venitian Blinds)

Típus: Bump

A textúra egy bump map sorozatot helyez a tárgyra, amittől az úgy fog kinézni, mint egy redőny, például a relaxa. A redőny elemei a textúra Z-X síkján fekszenek, X-el párhuzamosan.

Paraméterek:

**Z Size** - Az redőnylapok magassága.

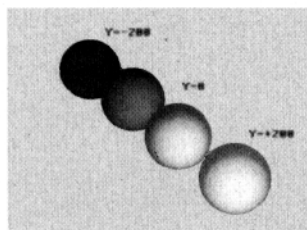
**Bump Adjust** - A redőny elemeinek magassága, vagy negatív érték esetén mélysége.



### ZBUFFER

Típus: Color

A ZBuffer igen különleges textúra, attól függően változtatja a tárgyak színét, hogy azok a globális Y tengely mentén hol helyezkednek el. Ha a tárgy mozog a globális Y tengely mentén, a tárgy színe ennek megfelelően változik. Két színt adhatunk meg, ezek lesznek a határszínek, valamint megadhatjuk azt is, hogy ezek a színek a globális Y tengely mentén hol helyezkednek el. Ha a tárgy túlhalad e két határon, színe felveszi a megfelelő határértéket és nem változik tovább.



Paraméterek:

- Y1, Y2** - Az első és hátsó határok Y koordinátái. Ezeket elérve, a tárgy színe felveszi a neki megfelelő határszint, túljutva ezen tovább már nem változik.
- Y1, Y2 Color** - Az Y1 és Y2 koordinátákhoz tartozó határszínek.

## Animálható textúrák

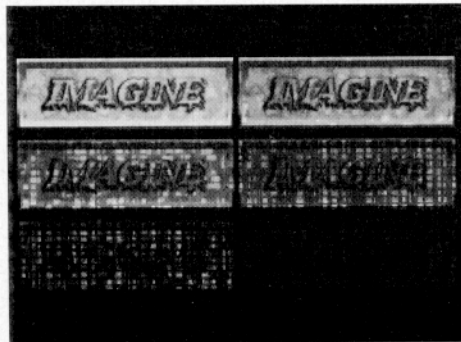
Ebbe a csoportba olyan textúrák tartoznak, amelyek rendelkeznek egy időváltozóval, amellyel a textúra által kialakított mintázat az animáció során változtatható. Ez tehát nem azt jelenti, hogy a többi textúra a paramétereik változtatásával nem lenne animálható, hanem azt, hogy ezek a textúrák speciálisan erre a feladatra vannak felkészítve. Jellemző ezekre a mintázatokra, hogy ha a mintázat ismételhető, akkor az időváltozó kezdeti 0.0 és végső 1.0 értékeinél a mintázat azonos, elősegítve a zökkenőmentes loopolást. Használatuk a State-k segítségével történik, legalább két state-t kell létrehozni, ezekben az időváltozót módosítva létrejön az animáció.

### BEAMMEUP

Típus Color, Filter

A textúra a StarTrek filmekből jól ismert Beam-out effektus utánczására szolgál. A tárgy miközben elhalványodik, színes részecskékre bomlik szét.

A részecskék színe, sűrűsége és a közöttük lévő távolság a textúra kérdezőiben beállítható. A hatás





## Textúrák

---

csak akkor lesz tökéletes, ha a tárgy az animáció végére nulla Specular és Hardness attribútumokkal rendelkezik, ugyan is a részecskékre bomlás az átlátszóság változtatásával jön létre. Ha a teljesen átlátszó, ezért nem látható részek becsillannak, az furcsán néz ki.

Paraméterek:

**Particle Size** - A tárgy részecskéinek mérete.

**Fraction 'On'** - A részecskébe alakuló felület aránya a teljes felületről. A részecskévé nem alakuló részekből lesznek a részecskék közötti üres területek. Ha ennek a paraméternek az értéke 1.0, az egész felület részecskékké alakul, nem lesz közöttük üres terület, nem is látszik, hogy a tárgy szétbomlott.

**Dist Traveled** - Az animáció időváltozója. Ennek az értéke mutatja meg, hogy az animáció a teljes cikluson belül éppen hol jár, ezt az értéket kell az animáció során változtatni. Lehetséges értéke a 0.0..1.0 határok között van.

**Min/Max Spacing** - A részecskék közötti távolság határai.

**Particle Rotations** - Az animáció során a részecskék foroghatnak a textúra tengelye körül. Ez az érték azt mutatja meg, hogy az animáció során a részecskék hány fordulatot tesznek meg.

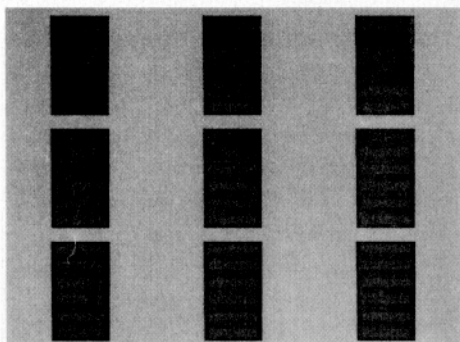
**End Filter** - A tárgy átlátszósága az animáció végén.

**Color 1, 2** - A részecskék két bázisszíne.

**CHASER**

Típus: Color, Reflect,  
Filter

A textúrával egy sornyi ablakot rajzolhatunk a tárgyra, mint például a toronyházak, vagy egy úrhajó ablakai. Az ablakok bekapcsolt, vagy



kikapcsolt Bright attribútumúak. A paraméterek változtatásával a textúra animációban is felhasználható, kialakítva vele LED kivezérlésjelzőt, leszállásvezérlő fénysort, stb. A textúra csak a tengelyének befoglaló keretén belül fejt ki hatását, a Z tengely határozza meg az alsó és felső részt, az X tengely pedig a jobb és bal oldalt.

Paraméterek:

**Number of Windows** - A textúra tengelyén belül elhelyezkedő ablakok, LED-ek száma.

**Window Size (0..1)** - Az ablakok mérete.

**Window Color** - Az ablakok színe.

**Window Bright (0..1)** - Brightness On/Off. Az ablakok Bright attribútuma, csak 0 és 1 értékek vannak értelmezve, a Bright ki-vagy be van kapcsolva.

**Window Fil/Ref Adj (0..1)** - Az ablakok fényvisszaverő és fényáteresztő képességének mértéke.

**Flat X/Wrap Z (0/1)** - A mapping eljárás típusa, sík, vagy hengeres.

**Shape Rect/Round (0/1)** - Az ablakok alakja, négyzetes, vagy kerek.

**Percent On (0..100)** - A felrajzolt ablakok százalékos aránya. Ennyi LED világít a kijelzőn.

## Textúrák

---

**On Ordered/Rnd (0/1)** - Azt szabályozza, hogy a bekapcsolt ablakok egymás mellett, összefüggően helyezkednek-e el, vagy össze-vissza.

**On Ránd Seed (0..1)** - Véletlen szám, amely a bekapcsolt LED-ek közötti helyek eloszlását szabályozza, ha nem összefüggők az aktív elemek.

**Anim Chase/Rnd (0/1)** - Az animációs mód, követő, vagy véletlen. Előbbi esetben a következő képen aktív LED-ek száma az előzőleg aktívakból következik, annak módosulata, utóbbi esetben ezek függetlenek egymástól.

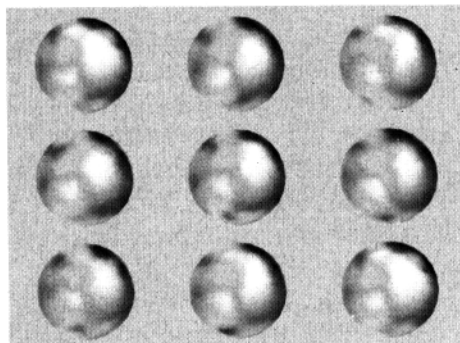
**Time Adj** - Az animáció időváltozója.

**Intensity Adj** - A textúra intenzitása, az eredeti attribútumokat lefedő képessége.

### CLOUDS

Típus: Anim, Color, Filter

A textúra elsősorban felhős ég utánzására alkalmas. Jelentős előnye a brush mapping-gal szemben, hogy a mintázat animálható, a felhők úszhatnak az



égen, változhat az alakjuk. A mintázat kérdezőjében két szín és átlátszóság értéket adhatunk meg, egyet a felhőknek, egyet pedig az égnek. A felhők mennyisége szabályozható. Egy horizont távolságot is megadhatunk, amely távolságban a felhők összeolvadni látszanak a horizonttal. A felhők átlagos mérete és elhelyezkedése a textúra tengelyével szabályozható.

Paraméterek:

**Density Adjust** - A felhők mennyisége a teljes égfelülethez képest.

**Horizon Fade Distance** - Ezzel a paraméterrel megszüntethetjük a groundra feszített textúra végtelen távolságban jelentkező hibáját. A program a képszámítás során csak eddig a word koordinátaig készíti el a groundra feszített felhőmintázatot. A koordináták a három tengely mentén azonosak, mindig a világ tengelyrendszerében értendők, függetlenül a tárgyak és a kamera pozíciójától. A korlátozó funkció kikapcsolható, ha ennek a paraméternek 0 értéket adunk.

**Cloud Color/Filter** - A felhők attribútumai. Negatív érték esetén azonos lesz a tárgy színével.

**Sky Color/Filter** - Az ég attribútumai. Negatív érték esetén azonos lesz a tárgy színével.

**Time Adj** - Az animált textúra időváltozója.

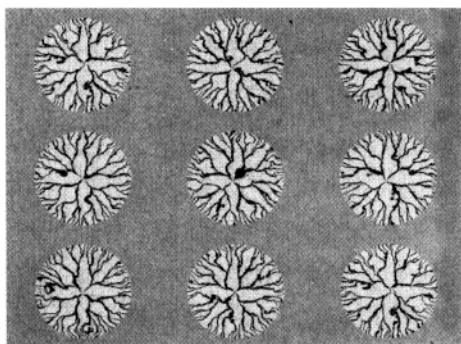
**Intensity Adjust** - A textúra attribútumainak fedő képessége. Alacsonyabb érték esetén a tárgy eredeti színe átlátszik a felhőkön és az égen. animáció során lecsökkentve, a felhők eloszlanak az égről.

### DANCSPRK

(Dancing Sparks)

Típus: Color, Reflect  
Filter, Bump

Ez a textúra vibráló plazmakisülést hoz létre. A művelet alapjául ugyan az az algoritmus szolgál, mint a Branches textúrának, a hatás tehát hasonló. A textúra rendelkezik egy animálható zaj paraméterrel és a plazma attribútumai is megadhatók. A kisülés szélessége, elágazásainak száma és bump mértéke szintén paraméterezhető. A plazmakisülés az X-Y



## Textúrák

---

síkon jön létre, a textúra Z tengelyének tehát kifelé kell mutatni a tárgyból.

Paraméterek:

**Branches (2, 4, 8 ...)** - A textúra középpontjából szétágazó ágak száma.

**Branch Width (0..1)** - Az ágak szélessége a teljes felülethez képest. Az 1.0 érték azt jelenti, hogy az ágak a teljes felületet lefedik, nem is lesz közöttük üres terület.

**Branch Clr/Ref/Flt** - Az ágak attribútumai. Negatív érték használatával az adott paraméter azonos lesz a tárgy eredeti attribútumával.

**Dist Traveled** - Az animációs folyamat helyzete, vagyis az időváltozó.

**Travel Magnitude Adj** - Az animálható noise magnitude paraméter. A szerteágazó ágak csavarodásának a mértéke.

**Noise 1, 2** - Az ágak összevisszaságát kialakító zaj paraméterei.

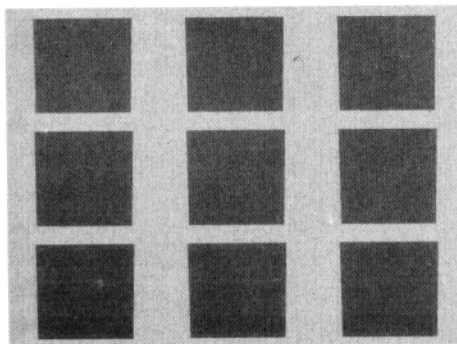
**Bump Adjust** - A plazmaágak kiemelkedése, vagy besüppedése.

### DRIPDROP

Típus: Bump

Ez olyan hatást vált ki a tárgy felületén, mint a vízbe csöppenő esőcsepp. A hullámok hullámhossza, amplitúdója sugara és a hullámok száma paraméterez-

hető, a hullám kezdeti és végső állapota megadható. Az animáció során a hullám a kezdőértékből átalakul a végső állapotba. A hullámok az X-Y síkon jönnek létre.



Paraméterek:

**Start/End Wavelength** - A hullámok kezdeti és végső szélessége.

**Start/End Z Amplitude** - Hasonló a bumhoz, a hullámok kezdeti és végső magassága/mélysége.

**Start/End Max Radius** - A hullámok által keltett legnagyobb gyűrű sugara.

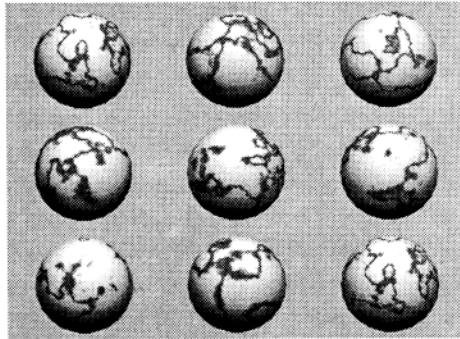
**Start/End Ripple Count** - Ennyi hullámgyűrűből áll egy-egy hullám.

**Dist Traveled** - Az animáció időváltozója. A kezdeti 0.0 értékénél a Start paraméterek jellemzik a hullámot, míg a végső 1.0 értékénél az End paraméterek.

### ELECTRIC

Típus: Anim, Color, Reflect, Filter, Fog

A textúra véletlen rendben kigyózó indákat visz a tárgyra, ehhez módosíthatja a színt, fényvisszaverő, fényáteresztő, képessé-



get, vagy a kód attribútumot. A textúra tengelyét animációban mozgatva hatása olyan, mintha a tárgy felszínén elektromos kisülés haladna végig. Nem csak a minták pozíciója, hanem a mérete is a textúra tengelyével szabályozható.

Paraméterek:

**Color/Filter/Reflect** - A textúra által keltett indák attribútumai. Az indák széle lágy átmenettel illeszkedhet a tárgyhoz.

**Width Adj** - A mintázat szélessége.

## Textúrák

---

**Turbulence Adjust** - A mintázat turbulenciája. nagyobb érték kuszább mintázatot eredményez, de a rendering is lassul.

**Noise 1, 2** - A mintázatot kialakító fraktál zaj paraméterei.

**Distance Traveled** - Az animáció időváltozója. A kialakuló mintázat ismétlődik, jól loopolható.

**Intensity Adjust** - A textúra intenzitása, vagyis az, hogy mennyire fedje le a már meglévő attribútumokat.

### FIRE

Típus: Anim, Color, Filter

A textúra a tárgy átlátszóságának változtatásával tűz hatását próbálja kelteni. A tűz színe és lobogásának sebessége szabályozható.

A tű a textúra tengelyein belül ég, a lángok +X és -X irányban lobognak

Paraméterek:

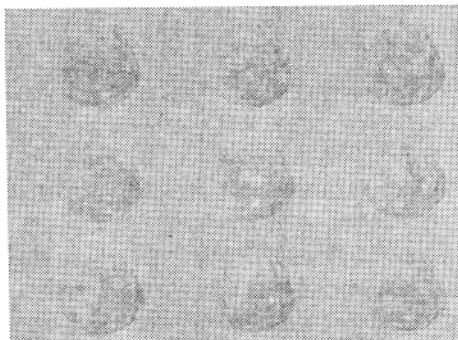
**Fragmentation Adjust** - A lángok töredezettsége, rendezettsége. Minél nagyobb ez az érték, a tűz annál szabálytalanabb, pattogóbb lesz, kis érték esetén inkább fog hasonlítani a gyertya lángjához.

**Convection Speed** - A lángok sebessége az animáció során.

**Central Color/Filter** - A tűz közepének attribútumai.

**Outer Color/Filter** - A tűz széleinek színe.

**Time Adjust** - Az animációs időváltozó. Sajnos a textúra animációja nem loopolható, a ciklus újraelkezdésekor ugri a mintázat.

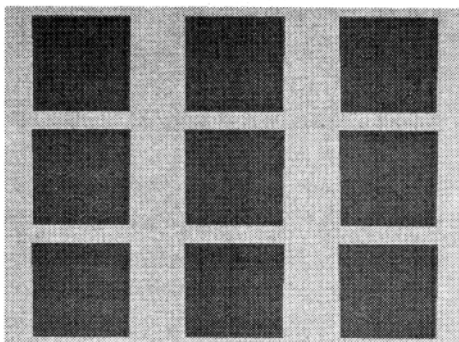


**Intensity Adjust** - A textúra intenzitása, fedőképessége. Alacsonyabb érték esetén a tárgy attribútumai átlátszanak a tűzön. Értékének változtatásával a tűz kihunyhat, vagy felerősödhet.

### RAIN

Típus: Bump

A textúra esőáztatta vízfelszín hatását kelti, alapjában hasonló a DripDrop textúrához. A hullámhossz, amplitúdó, rádiusz és az egyszerre látható hullámok száma paramétrezhető. Az esőcseppek ismétlődő mintázat szerint esnek, az időváltozó 1.0 értéke után jövő 0.0 érték zökkenőmentesen illeszkedő mintázatot hoz létre. A textúra hatása csak a tengelyének befoglaló keretén belül jelentkezik. A hullámok az X-Y síkon jelennek meg, a tárgy Z tengelyének tehát kifelé kell mutatni a tárgy felületéből.



Paraméterek:

**Min/Max Wavelenght** - A hullámhossz szélső értékei.

**Min/Max Z Altitude** - A hullámmagasság szélső értékei.

**Min/Max Radius** - A hullámok sugarának szélső értékei.

**Min/Max Ripple Count** - Egy-egy hullámhoz tartozó hullámgyűrűk számának szélső értékei.

**Min/Max Lifetime** - A hullámok élettartama a teljes ciklushoz képest. Például ha ennek értéke 0.5, akkor egy hullám a Distance Travel értékének 0.5-tel való változása alatt jön létre és szűnik meg.



## Textúrák

---

**Dist Traveled** - Az animáció időváltozója, ennek 0.0-ról 1.0-ra való változása során jön létre egy animációs ciklus. Az animáció ismételhető, az 1.0 után zökkenőmentesen következik a 0.0, a ciklus loopolható.

**# of Drops in Cycle** - Egy animációs ciklus alatt létrejövő hullámsorok száma.

**Random Seed** - Véletlen szám az effektus egyénibbé tételéhez.

### SPARK

Típus: Color, Reflect, Filter

Ez egy elektromos elektródák között létrejövő, vibráló, ugráló szikraözönt szimulál. Az animáció loopolható, zökkenőmentesen is-

mételhető. A szikraözön a tárgy X-Y síkján jön létre, tehát a Z tengelynek kifelé kell mutatni a felületből.

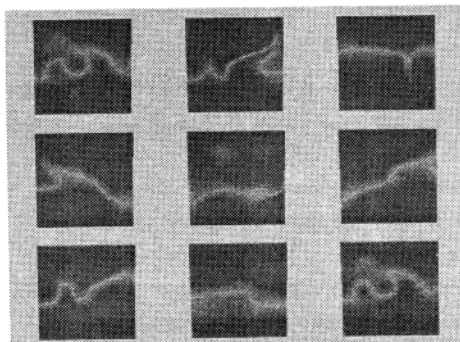
Paraméterek:

**Dist Traveled** - Az animációs ciklus pozíciómutatója.

**Travel Magnitude Adj** - Animálható zaj magnitúdó paraméter. Ezzel változtatható a szikrasor csavarodottsága.

**Color/Reflect/Filter** - A szikrák attribútumai. Negatív érték hatására az adott attribútum azonos lesz a tárgy eredeti attribútumával.

**Noise 1, 2** - A szikrák alakjának formálását végző fraktál zaj paraméterei.

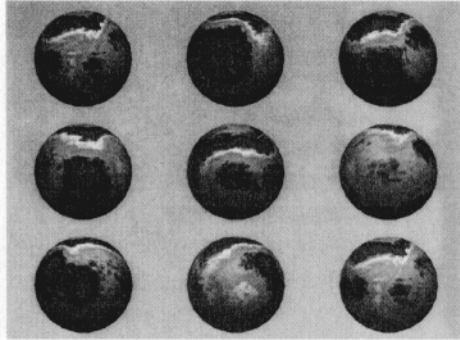


**TIEDYE (Tie Dyed**

Rings)

Típus: Color

Ez a textúra három bázisszín véletlenszerű körkörös keverését hozza létre a textúra Z tengelye körül, mint az egymásra öntött különböző színű festékek.



A keveredés zaj alapú algorit-mussal animálható, valóságos színorgiát hozva létre. Gömbön alkalmazva jól megválasztott színek használata esetén olyan hatást kelthetünk, mint egy idegen bolygó légköréről készült hamis színezésű műholdfelvétel, amelyen látható a légkör mozgása. A foltok mérete és elhelyezkedése a textúra tengelyével szabályozható.

Paraméterek:

**Noise 1, 2** - A festékfoltok alakjának változatossága.

**Color 1, 2, 3** - A festékfoltok bázisszínei. Negatív érték használata esetén az adott szín azonos lesz a tárgy meglévő színével.

**Travel Magnitude** - Animálható zaj paraméter, a színek keveredésének mértékére van hatással. Minél nagyobb, annál inkább egymásba vannak kavarodva.

**Dist Traveled** - A textúra animációjának időváltozója. A mintázat ismétlődő, loopolható.

### Köd textúrák

Ebbe a csoportba ködszerű attribútumot kialakító textúrák tartoznak. Jelentős különbség ezek és a normál Fog attribútum között, hogy nem csak azonos intenzitásúak lehetnek a teljes felületen, hanem gomolygó, vagy egyéb hatásokat is kialakíthatunk. Ezeket a textúrákat köd attribútummal rendelkező tárgyakon kell alkalmazni, általában a már meglévő köd tulajdonságot alakítják tovább.

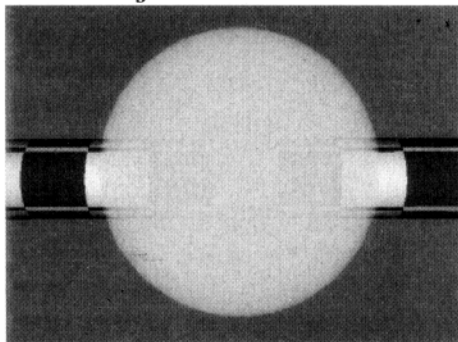
#### GHOST

Típus: Fog

Ez a textúra megváltoztatja a köd sűrűségét a tárgy éleinél, ezáltal anyagtalán kinézetet adva annak. A textúrát fog objecten kell alkalmazni. A köd színe módosítható attribútummal brush-al, vagy akár textúrával, létrehozva ezzel fénysugár, vagy gázláng kinézetű tárgyakat.

Paraméterek:

**Fog Lenght at T** - A köd sűrűsége egy köztes részen. A kamerával szemben lévő része a tárgynak megfelel a 0.0 értéknek, míg az erre merőleges oldalsó rész 1.0-nak. A 0.0 részen, vagyis a kamerával szemben a tárgy attribútumai között megadott fog lenght érvényesül, míg az 1.0 területen, a kamera tengelyére merőlegesen egy fix érték, amely gondoskodik arról, hogy tárgy széle teljesen köddé váljon. E két határ között megadhatunk egy részt a következő input mezőben, ahol ez lesz a köd sűrűsége, ezáltal a tárgyon lévő köd attribútum átmenete szabályozható.

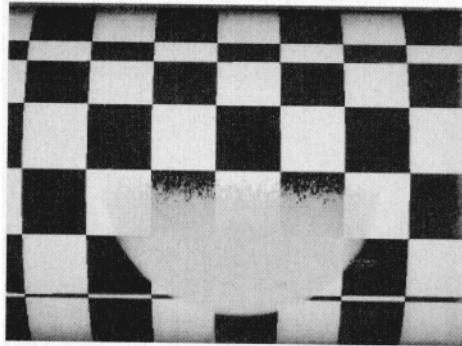


**T** - Ez egy treshold érték, amely azt mutatja meg, hogy a tárgy mely részén érvényes az előző input mezőben megadott fog lenght. A könnyebb érthetőség miatt nézzünk erre egy példát. Tegyük fel, hogy a tárgy attribútumai között megadott fog lenght értéke 10, a Fog Lenght at T pedig 20. Ha T értékének 0.5-öt adunk, akkor a tárgy felületének az előbbi szemlélet szerint 0.5 értéknél lévő része, vagyis kb. a kamera tengelyével  $45^\circ$ -ot bezáró felületrészek fog lenght-je 20 lesz. A kamerával szembeni 10-es értékről folyamatosan változik a kód sűrűsége eddig majd innen folytatja a változást a tárgy oldaláig, ahol eléri a fixen programozott értéket, ami a legjobb éleltűnést eredményezi.

### FOGTOP

Típus: Fog

Ez a textúra nagyon hasonló a MntnTop-hoz, de nem a tárgy színét változtatja, hanem a kód sűrűségét. Ez azt eredményezi, hogy a tárgy fokozatosan eltűnik a Z tengely



irányában. A textúrát fog objecten kell alkalmazni. A textúra méretezhető és pozícionálható a tengelye által.

Paraméterek:

**Fog Lenght at T** - A kód sűrűsége egy köztes részen. A textúra tengelypontjában lévő része a tárgynak megfelel a 0.0 értéknek, míg a Z hosszának megfelelő távolság 1.0-nak. A 0.0 részen a tárgy attribútumai között megadott fog lenght érvényesül, míg az 1.0 területen, a textúra Z tengely

## Textúrák

lyének csúcsán egy fix érték, amely gondoskodik arról, hogy tárgy teteje teljesen köddé váljon. E két határ között megadhatunk egy részt a következő input mezőben, ahol ez lesz a kód sűrűsége, ezáltal a tárgyon lévő kód attribútum átmenete szabályozható.

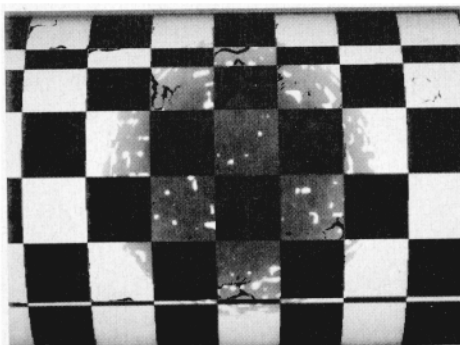
**T** - Treshold érték, amely azt mutatja meg, hogy a tárgy mely részén érvényes az előző input mezőben megadott fog lenght. A könnyebb érthetőség miatt nézzünk erre egy példát. Tegyük fel, hogy a tárgy attribútumai között megadott fog lenght értéke 10, a Fog Lenght at T pedig 20. Ha T értékének 0.5-öt adunk, akkor a tárgy felületének az előbbi szemlélet szerint 0.5 értéknél lévő része, vagyis a textúra Z tengelyének felénél lévő felületrészek fog lenght-je 20 lesz. A tengelypontban lévő 10-es értékről folyamatosan változik a kód sűrűsége eddig, majd innen folytatja a változást a textúra Z tengelyének csúcsáig, ahol eléri a fixen programozott értéket, ami a legjobb csúcstűnést eredményezi.

**Noise** - A kódátmenet egyenletességét, vagy szabálytalanságát befolyásoló fraktál zaj paraméterei.

### FOGPAINT

Típus: Fog

Ez a textúra a tárgy kialakult színének vörös összetevőjét veszi alapul a fog attribútum megállapításához. Minél magasabb a szín vörös tartalma, annál ritkább lesz a kód a tárgy azon részén. A textúrát



fog objecten kell használni. A textúra működéséhez felhasznált zaj a textúra tengelyével pozicionálható és méretezhető.

Paraméterek:

**Fog Lenght at T** - T értéke ez esetben a tárgy színének vörös összetevőjéből származik. Ahol a vörös szín értéke 0, ott T értéke is nulla, míg a teljes vörös intenzitású részeken  $(R=255)$  T értéke 1.0 lesz. A  $T=0.0$  részeken a kód értéke a tárgy eredeti kód attribútuma lesz, míg  $T=1.0$ -nél a kód olyan értéket vesz fel, ami a tárgy azon részének teljes eltűnését eredményezi. Ezzel a paraméterrel egy köztes vörös értékhez rendelhetünk ködsűrűséget.

**T** - Az előző input mezőben megadott ködsűrűség ekkora vörös intenzitásnál érvényesül. Például ha  $T=0.5$ , akkor az előző paraméter  $R=128$  vörös erősségű felületrészek fog lenght-je lesz.

**Noise** - A kódátmenet szabályossága, vagy szabálytalansága befolyásolható ezzel a zaj paraméterpárossal.

**Senser Color** - Ez a ritkább kód színe.

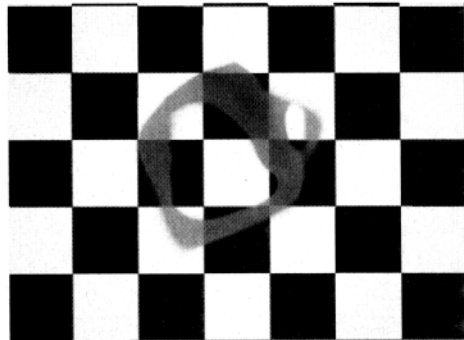
**2nd Color** - A kód színe a sűrűbb helyeken.

### NEBULA

Típus: Fog

Ez a textúra nagyon hasonló a Ghost-hoz, kivéve, hogy van egy Noise paramétere, amellyel zavarosabbá tehető a kód és néhány véletlen szín vihető a kódbe. A textúra azt a

hatás eredményezi, hogy a tárgy elhalványodik a szélei felé, a belsejében pedig gomolyogni látszik. A



## Textúrák

---

textúrát fog objecten kell használni. A textúra a tengelyén keresztül méretezhető és pozicionálható. Kiválóan alkalmas igazi gomolygó köd, vagy füst utánczására.

Paraméterek:

**Fog Lenght at T** - A köd sűrűsége egy köztes részen. A kamerával szemben lévő része a tárgynak megfelel a 0.0 értéknek, míg az erre merőlege oldalsó rész 1.0-nak. A 0.0 részen, vagyis a kamerával szemben a tárgy attribútumai között megadott fog lenght érvényesül, míg az 1.0 területen, a kamera tengelyére merőlegesen egy fix érték, amely gondoskodik arról, hogy tárgy széle teljesen köddé váljon. E két határ között megadhatunk egy részt a következő input mezőben, ahol ez lesz a köd sűrűsége, ezáltal a tárgyon lévő köd attribútum átmenete szabályozható.

**T** - Ez egy treshold érték, amely azt mutatja meg, hogy a tárgy mely részén érvényes az előző input mezőben megadott fog lenght. A könnyebb érthetőség miatt nézzünk erre egy példát. Tegyük fel, hogy a tárgy attribútumai között megadott fog lenght értéke 10, a Fog Lenght at T pedig 20. Ha T értékének 0.5-öt adunk, akkor a tárgy felületének az előbbi szemlélet szerint 0.5 értéknél lévő része, vagyis kb. a kamera tengelyével 45°-ot bezáró felületek fog lenght-je 20 lesz. A kamerával szembeni 10-es értékről folyamatosan változik a köd sűrűsége eddig. majd innen folytatja a változást a tárgy oldaláig, ahol eléri a fixen programozott értéket, ami a legjobb éleltűnést eredményezi.

**Noise** - A köd gomolygásának paraméterei.

**Central Color** - A köd sűrűbb részeinek színe. A ritkább részek színét a tárgy attribútuma határozza meg.

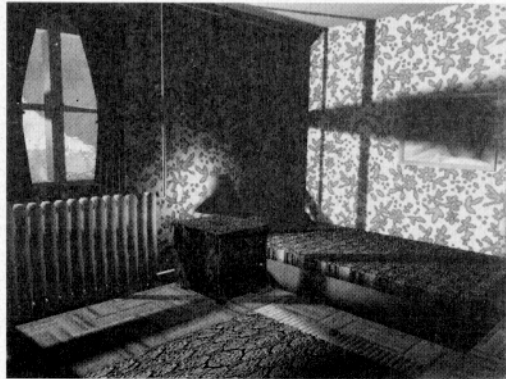
## Light textúrák

Ebbe a csoportba olyan textúrák tartoznak, amelyek nem normál tárgyra, hanem fényforrásokra és Light tárgyra lehet alkalmazni, velük megváltoztathatjuk a fény tulajdonságait, lágyíthatjuk, stb.

### FRNCHWIN (French windows Light)

Típus: Minden fényforrás

A textúra olyan hatást ad a fénynek, mintha az üvegezet ajtón jönne keresztül. Az üvegcellák közötti keretek árnyékot vetnek a fényben. A képzeletbeli ajtón lévő üvegcellák száma és az elválasztó keretek szélessége vízszintesen és függőlegesen külön-külön megadható.



Paraméterek:

**Num Vert/Horz Planes** - Az ajtón lévő üveglapok száma függőlegesen és vízszintesen.

**Vert/Horz Shade Adj** - Üvegcellák között lévő nem átlátszó függőleges és vízszintes keretek szélessége. Ezek a keretek árnyékot vetnek a fényben. A szélesség egy cella és a hozzá tartozó keret méretéhez képest értendő, tehát a 0.5 azt jelenti, hogy az üveg és a befoglaló kerete azonos szélességű lesz. Ha értéke 0.0, nem lesz keret, így a textúrának sem lesz hatása, ha 1.0, akkor a teljes ajtófelületet a keret teszi ki, vagyis tömör ajtó jön létre, amelyen nem hatol át fény.

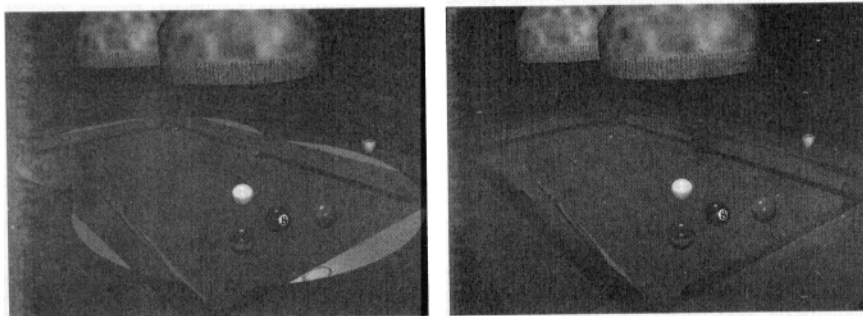


## Textúrák

---

**Penumbra Adj** - A fény félárnyékos szélének mérete az egész fénysugár méretéhez képest. Magasabb érték lágyabb fénysugarat eredményez.

### SOFTEDGE



Típus: Hengeres és négyzetes fényforrásokhoz.

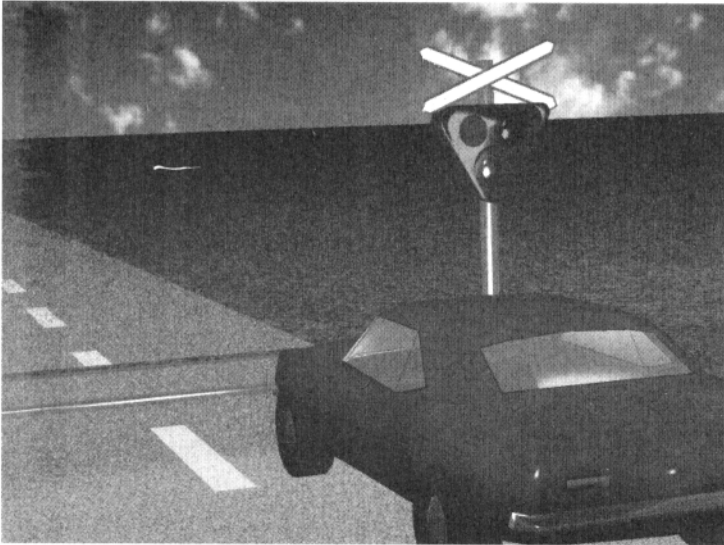
Az alakos fényforrások szélét lágyíthatjuk ezzel a textúrával, így nem lesz azoknak éles kontúrja, mint egyébként. A fenti képek közül a bal oldalin az asztalt normál kerek fénycsóvájú fényforrások világítják meg, míg a jobb oldalin ugyan ezek SoftEdge textúrával. A különbség jól látható.

Paraméterek:

**Penumbra Adj** - A fénysugár lágy szélének mérete az egész fénysugár méretéhez képest.

**Noise 1, 2** - A szórt fényű részbe vihetünk ezzel a két paraméterpárossal változatosságot, ezáltal nem lesz a fényforrás széle valószerűtlenül egyenletesen halványuló.

## STROBE



Típus: Minden fényforrás

Ez egy animált fény textúra, amely ciklikusan, ugrásszerűen változtatja a fényforrás színét két érték között., mint egy stroboszkóp fényforrás, jól utánozható vele például diszkók fényjátéka. Ha a fényforrás egyik színének 0,0,0-t adunk meg, a hatás olyan lesz, mintha az ki-be kapcsolódna, akár csak a vasúti átjárót biztosító lámpa.

Paraméterek:

**Strobe Cycles** - Annak mértéke, hogy hányszor alakuljon át ciklikusan a fényforrás színe a két érték között egyetlen animációs ciklus során. A zökkenőmentes loopolás miatt egész értéknek kell lennie, de elfogadott a tört szám is.

**Light Color 1, 2** - A fényforrás két szélső színe.

**Dist Travelled** - A textúra animációs időváltozója.

## VENLITE (Venetian Blind Lighting)



Típus: Minden fényforráshoz

A textúra olyan hatást kölcsönöz a fénynek, mint ha az lécrolettán, zsalugáteren jönne keresztül. A lécek az X-Y síkon X-el párhuzamosak

Paraméterek:

**Number of Shadows** - A redőnylécek száma, ennyi vízszintes árnyék jelenik meg a fényben.

**Shadow Sz Adj** - Az árnyék mérete, vagyis a lécek egymástól vett távolságának aránya a léc és egy hozzá tartozó rés méretéhez mérten. A 0.5-ös érték azt jelenti, hogy a lécek és a rések mérete azonos.

**Penumbra Adj** - A fények szélén lévő félárnyékos, lágy rész mérete.

**Noise 1, 2** - A fény lágy szélének egyenetlensége. Segítségével kiküszöbölhetők a túl szabályos, gépies körvonalak és szabályos átmenetek.

## Animációs effektusok

A program 3.2 változata jó néhány új animációs effektust tartalmaz, sőt, a régiek némelyike is átdolgozásra került. Az effektek leírásánál hasonló szerkezetet követtem, mint a textúrákénál, az effekt kincstári neve után egy rövid funkcionális leírás következik, majd az egyes paraméterek részletezése. Csak azoknak az effekteknek a leírása található meg a következő részben, amelyek a program 2.0-s változatában még nem voltak meg. A régiek leírása a kötet első részében olvasható. Kivételt ez alól csak a FireWorks effekt jelent, amelyet kissé kibővítettek, ezért leírását megismétlem.

Megjelent egy új effekt típus, a **Global Effect**, amelyet nem valamely tárgyra, hanem a Globals-ra kell alkalmazni, hatása az egész képen globálisan jelentkezik. Ilyen Global Effect például a Toon, amely az egész képet rajzfilmszerűvé teszi, vagy a nagyon divatos Lens Flare, amely a direkt kamerába világító fénynek a kamera lencséin történő csillanását utánozza. Ezeket az effektusokat külön fejezet-részben tárgyaljuk.

### Normál effektek

Normál tárgyra alkalmazható animációs effektusok, azokon fejtik ki hatásukat, a tárgy alakját módosítva.

#### ANIMBRSH

Igen egyszerű effektus, a tárgyra feszített **IFF-Animbrush**, vagy PC platformon **FLC** animáció felhasználását szabályozza. Kérdezőjében elérési úttal együtt meg kell adni a szabályozni kívánt

## Animációs effektusok

---

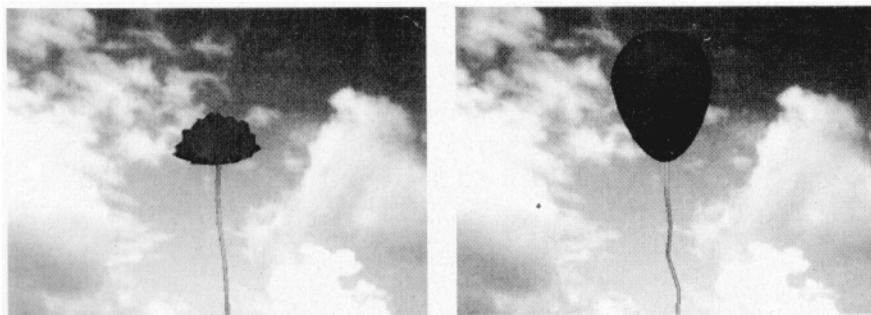
animbrush nevét, aminek szerepelnie kell a tárgyon is mint brush, egyéb beállításait ott kapja meg. Az effekt kérdezőjében mindössze azt adhatjuk meg, hogy az animációból mely kockák közötti rész kerüljön ciklikusan a tárgyra.

Paraméterek:

**Brush Name** - A kiválasztott brush neve, elérési úttal együtt.

**First, Last frame** - A felhasználni kívánt első és utolsó képkocka száma. Ha ez kevesebb képet határoz meg, mint a vele kapcsolatos animáció hossza, akkor a brush képei ciklikusan ismétlődni fognak.

### BALOON



Ez egy léggömbfújó effektus, segítségével a tárgyat felfújhatjuk, vagy leereszthetjük, mint egy léggömböt. A leeresztés során nem csak a tárgy mérete csökken, hanem a felülete is megroggyan, mint a valódi lufinak. Ezt úgy éri el a program, hogy a tárgy egyes pontjait véletlenszerű mértékben mozgatja a tárgy középpontja felé. Az, hogy a lufi leereszt, vagy felfújódik, attól függ, hogy a tárgynak mekkora a tengelye. Ha a tengely belül esik a tárgyon, a pontok befelé mozdulnak, a lufi leereszt, ha kívül esik, ak-

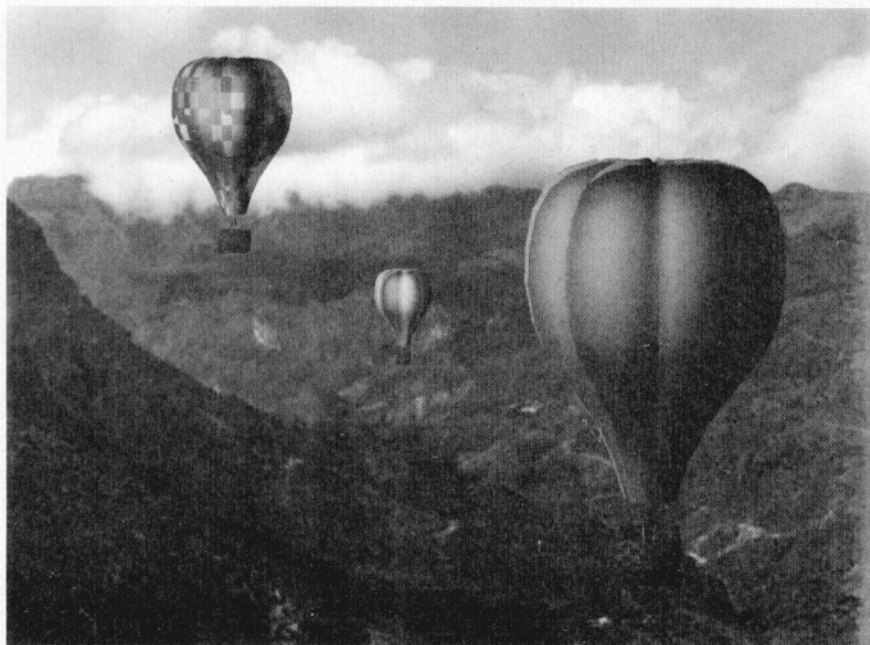
## Animációs effektusok

kor kifelé mozdulnak a pontok, a lufi felfúvódik. Kivételt jelent ez alól, amikor egy megadott méretűre változtatjuk a tárgyat.

Paraméterek:

**Inscribe within obj** - Ez és a következő három kapcsoló annak kiválasztására szolgál, hogy a lufi hogyan változtassa alakját. Egyszerre csak egyet használhatunk közülük. Ha ezt ikszeljük be, akkor a tárgy pontjai jellemzően beesnek a tárgyba, az összefonnyadni látszik, még ha a mérete a fentebb elmondottak alapján növekszik is.

**Circumscribe about obj** - A tárgy felületén dudorok jelennek meg, a látvány ellentétes az előzővel, függetlenül, hogy a méret hogyan változik.



## Animációs effektusok

---

**Use Average Radius** - Egyenlő mértékben keletkeznek beesett és kidudorodott egyenetlenségek a tárgy alakján.

**Set Radius to** - A tárgy felületének nem változik az alakja, csak a mérete alakul akkorára, hogy a befoglaló kerete az input mezőben megadott gömbön belül épp elférjen. Hogy ez a változás méretcsökkenés, vagy növekedés, a tárgy eredeti méretétől függ.

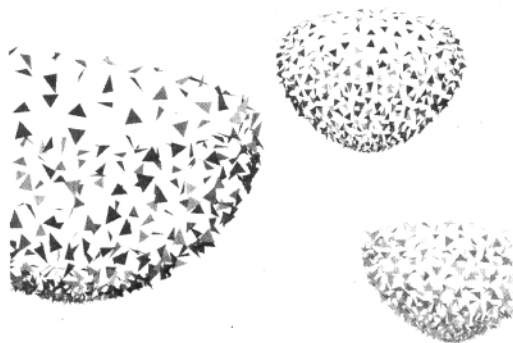
**Reverse Baloon Timing** - Bekapcsolva a lufi méretváltozása fordítva megy végbe. Például ha egy 100 egység sugarú gömböt 50 egység sugarúra eresztünk le, de ez az opció be van kapcsolva, akkor a gömb az animáció kezdetén lesz 50 egység sugarú és innen fújódik fel 100 egység sugarúra.

**Return to Original Shape** - Az animáció során a tárgy alakja az effekt hatásának végére visszaáll eredeti formájára. Például a lufi méretét csökkentve, az csak az animációs ciklus feléig csökken, utána növekszik.

### FIREWRKS

Az effektus megtalálható volt már az Imagine 2.0-ban is, de azóta kissé kibővült, ezért itt újból megvizsgáljuk. Ezzel felrobbanthatjuk a tárgyat alkotó elemeire, azaz önálló felületrészekre, amelyekre Z gravitáció is

hat, ezért a szétrobbanó darabok visszahullnak. A robbanás során az egyes darabok véletlenszerű



## Animációs effektusok

---

forgást végezhetnek, méretük megváltozhat. Az effekt hatása azonnal jelentkezik, már az első képen szétszabdalja a tárgyat, amelyiken alkalmazva lett.

Paraméterek:

**Spherical/Radial/Linear** - Három kapcsoló, melyek közül egyszerre egy lehet aktív. Ezekkel választható ki a robbanás iránya. Spherical esetén a darabok gömbszerűen, Radial esetén a tárgy egyik tengelye körül körkörösén, míg a Linear-t választva egy tengellyel párhuzamosan röppenek szét.

**X Axis/Y Axis/Z Axis** - A robbanás tengelyét kiválasztani rendeltetett kapcsolók. Gömbszerű robbantás esetén nincs jelentőségük, hiszen ekkor az elemek mindhárom tengely mentén egyformán széttartanak, de Radial, vagy Linear választása esetén ezen tengely körül, illetve mentén röppennek szét a darabok.

**Explosion distance** - Az a távolság, amelyre a tárgy darabjai eltávolodnak a tárgy középpontjától a robbanás során.

**Expansion Angle** - A darabok emelkedési szöge a robbanás aktív szakaszában.

**Triangle Scaling** - Az egyes darabok méretének változása az effekt működése során. Az input mező értéke az elemek végső és eredeti méretének aránya.

**Minimum/Maximum # of Rotation** - A szétröppenő darabok forgást is végeznek, ha e két input mező közül legalább egynek nem nulla az értéke. A minimális és maximális értéket nem szögfokban, hanem fordulatban kell megadni, de ez lehet törtszám is.



## Animációs effektusok

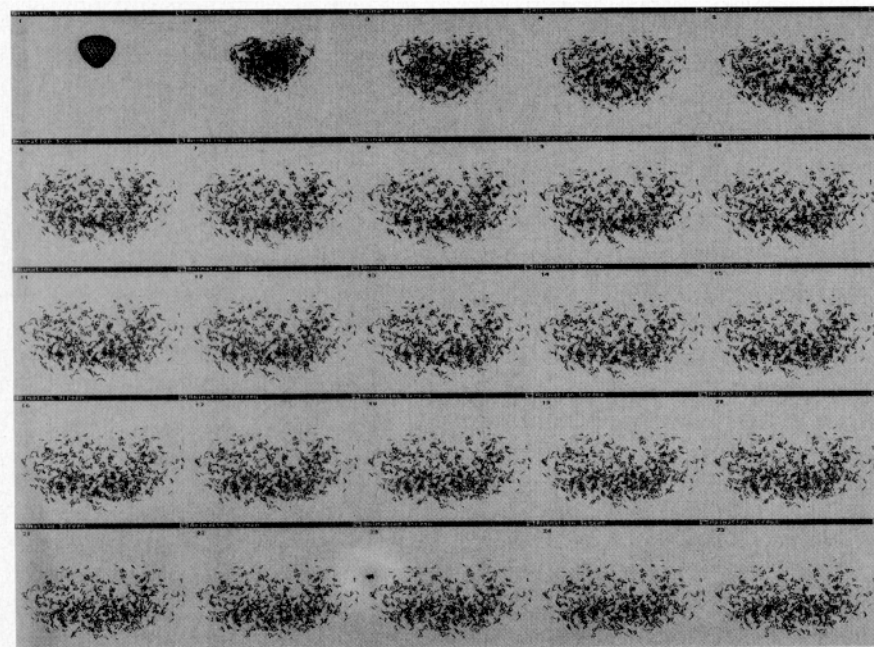
**Frame Count for Expansion** - Azon képkockák száma, amelyen keresztül a darabok emelkednek és erőteljesen távolodnak egymástól. A robbanás középpontja a tárgy tengelye, ezért az ez alatti elemek nem emelkednek, hanem lefelé robbannak, de a fő tágulás ezeknél is a megadott kockák alatt megy végbe.

**Distance to fall by end** - Az a távolság, amelyen a szétröppenő darabok az effekt végére az eredeti magasságukra visszaesnek.

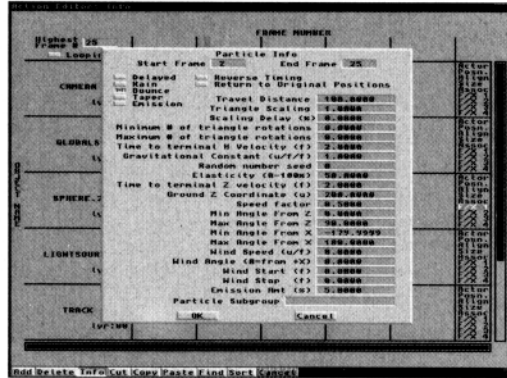
**Random number seed** - A robbanás kialakításához használt véletlen szám, értéke bármi lehet.

**Make faces sparkle** - Ezt a kapcsolót kiikszelve a szétröppenő darabok szikrázó hatást keltenek.

### PARTICLE



Az Imagine legbo-nyolultabb effektusa, segítségével a tárgyat elemeire robbanthatjuk szét, hasonlóan a Explode, vagy Firewrks effektushoz, de jóval több lehetőséget vehetünk figyelembe. Például a darabokra szél hathat, megadott Z magasságon lévő X-Y síkra visszaesve pattoghatnak, mintha a talajra estek volna, stb.



Paraméterek:

**Delayed** - Ha aktiváljuk, az effekt hatása nem azonnal, hanem csak a második képkockán kezdődik, az első kockán a tárgy még érintetlen marad.

**Rain** - Ha ezt az opciót használjuk, az effektből kimarad a robbanás táguló része, helyette a tárgy végig kitölti a travel distance által meghatározott méretet, de az elemek folyamatosan cirkulálva távolodnak az eredeti helyzetükből a kellő irányba. Az eltávolodó elemek a megadott magasságban lévő talajszintet elérve eltűnnek, de eredeti pozíciójukból újra indulnak. Az effekt hatása ekkor olyan, mintha szökőkútból, vagy esőfelhőből ömlene a víz.

**Bounce** - Ezt az opciót kiválasztva a lehulló darabok csak egy megadott magasságon elhelyezkedő X-Y síkig esnek vissza, majd azon pattogni kezdenek az elasztikusságuktól függő mértékben. A tárgy azon részei, amelyek már a robbanás előtt e sík alatt helyezkedtek el, háborítatlanul tovább zuhannak

## Animációs effektusok

---

**Taper** - Ha ezt a kapcsolót aktiváljuk, a tárgy a robbanás során nem gömbszerűen tágul, hanem +Z irányában kicsúcsosodik.

**Emission** - Kapcsoló, melyet aktiválva nem az egész tárgy robban fel, hanem csak darabok röppennek ki belőle. Ezek a darabok ugyan úgy viselkednek, mintha a tárgy egésze felrobbant volna. A Delayed-del és a Rain-nal együtt nem használható.

**Reverse timing** - Az effekt időbeni irányát meghatározó kapcsoló. Aktiválása után az effektus hatása fordított lesz.

**Return to original positions** - Ping-pong hatást létrehozó opció, a kapcsolót aktiválva az animáció oda-vissza megy végbe, a befejezésre a tárgy visszanyeri eredeti alakját.

**Travel distance** - Az a távolság, amelyre a tárgy elemei eltávolodnak az effektus működése során.

**Triangle scaling** - Az egyes darabok méretváltozásának aránya az effekt végére.

**Scaling delay (%)** - Az elemek méretváltozásának kezdete az effekt teljes hosszához képest. Ha pl. értéke 50, az elemek csak az effekt felétől kezdik méretüket változtatni.

**Minimum/Maximum # of triangle rotations** - A szétrobbanó darabok minimális és maximális elfordulásának száma. Az egyes darabok valós elfordulása e két érték között véletlenül választódik ki. Ha mindkét érték nulla, a darabok nem fordulnak el. Az elfordulást nem szögfokban, hanem fordulat-számban kell megadni, törtszám lehetséges.

**Time to terminal H velocity (f)** - Azon képkockák száma, amelyek alatt gyorsulva a szétrobbanó darabok legnagyobb vízszintes sebességüket elérik.

**Gravitational constant (u/f/f)** - Gravitációs állandó Imagine egység/frame<sup>2</sup> egységben.

## Animációs effektusok

---

**Random number seed** - Véletlen szám az effekt működésének változatosabbá tételére. Értéke bármilyen szám lehet.

**Elasticity (0-100%)** - Az egyes darabok rugalmassága. Mozgási energiájuk ekkora része marad meg egy-egy ütközés után.

**Time to terminal Z velocity (f)** - Azon képkockák száma, amelyek alatt gyorsulva a szétrobbanó darabok legnagyobb függőleges sebességüket elérik.

**Ground Z coordinate (u)** - A képzeletbeli talaj Z pozíciója, amelyre a darabok visszaesnek, esetleg ezen pattognak.

**Speed factor** - A darabok röptének sebességtényezője.

**Min/Max angle from Z** - A szétröppenő elemek függőleges irányszögeinek minimális és maximális értéke a Z tengelyhez viszonyítva. A 0-90 értékek azt jelentik, hogy a darabok a felfelé tartó függőlegestől a vízszintesig bármilyen szög alatt repülhetnek. Ha pl. mindkét értéknek 40-et adunk, a darabok egy szűk sávban, 40°-os szög alatt emelkednek, a 90-180 értékpáros hatására pedig a darabok a vízszintes és a lefelé tartó függőleges irányok között szállnak.

**Min/Max angle from X** - A darabok szóródása a tárgy Z tengelye körül. A 0° a +X irányába esik. Az alap 180;-180 értékpáros azt eredményezi, hogy az elemek a Z tengely körül körben egyenletesen szóródnak szét. A 0; 180 értékpáros hatására az elemek az X tengelytől jobbra, +Y irányában félkörben szóródnak.

**Wind speed (u/f)** - Az effektus során egy oldalirányú szél is hathat a darabokra, ez az érték a szél a sebessége Imagine egység/képkocka mértékegységben.

## Animációs effektusok

---

**Wind angle (0=from +X)** - A szél iránya fokokban.

Az észak +X-nél van, tehát a 0 érték azt jelenti, hogy a szél +X irányából fúj, a 180 pedig azt, hogy -X felől.

**Wind start/stop (f)** - A szél hatásának kezdete és vége, képkockában megadva. Ezek az értékek az effekt kezdetéhez relatívak, vagyis nem az animáción belüli hanem az effekt kezdetétől számított képkockák számát jelentik.

**Emission amt** - Ha Emission robbantási módot választunk a kérdező bal felső sarkában lévő kapcsolókkal, akkor ennek az input mezőnek az értéke határozza meg az egyszerre kibocsátott felületelemek mennyiségének arányát a tárgy teljes felületéhez képest.

**Particle subgroup** - Nem csak az egész tárgyat robbanthatjuk fel, hanem akár csak annak egy részét is. A robbantandó részt subgroupnak kell meghatározni, ennek a subgroupnak a nevét kell ide írni. Ha nincs írva semmi, az effekt az egész tárgyra vonatkozik.

### SHREDDER

Ez egy igen hatásos variáció a robbanás témára. A tárgyakat nem csak elemi felületrészekre képes szétszabdálni, hanem véletlen méretű és alakú darabokra is, sőt, akár képes a subgroupokat egyben kezelni. A robbanás során gravitációs tényezőt is megadhatunk, a visszahulló darabok egy



megadott Z magasságon fekvő X-Y síkra visszaesve pattoghatnak az elasztikusságuktól függő mértékben. Az effektus nagyrészt valódi mértékegységeket használ, a távolságot és az időt méterben és másodpercben adhatjuk meg. Ez úgy lehetséges, hogy egy input mezőben megadjuk az Imagine egység és a méter egymáshoz való viszonyát, egy másodperc pedig 25 képkocka.

Paraméterek:

**Whole Objects as Parts (Grouped Object)** - Ha ezt a kapcsolót aktiváljuk, a robbanás csak a csoportkapcsolatokra lesz hatással, a csoport elemei egyben maradnak, úgy távolodnak el egymástól.

**Use Defined Subgroups as Parts** - Kapcsoló, melyet aktiválva a tárgy subgroupjai egyben maradnak a robbanás során. Az egyetlen subgrouphoz sem tartozó részek az egyéb beállításoknak megfelelően robbannak szét.

**La Machine** - Általános robbantási mód, a csoportkapcsolatoktól és a subgroupoktól független véletlen méretű és alakú darabokra robban a tárgy.

**Maximum Number of Triangles in Group** - Legfeljebb ennyi felületelemből álló darabokra szakad a tárgy La Machine esetén, vagy Use Defined... választásakor a subgroupokhoz nem tartozó részek. Az egyes darabok alakja és mérete véletlenül választódik meg. Ez a mód lényegesen lassabb, mint az előző kettő.

**Bounding Subgroup** - A robbanást lekorlátozhatjuk a tárgy egy részére, ha ide a szétrobbantani kívánt subgroup nevét írjuk. A tárgy többi részére az effekt nem lesz hatással.

**Start/End Time (seconds)** - Az animáció irányának és sebességének megadását szolgáló input mezők. A Start az animáció kezdete, az End a vége, valós időben kifejezve. A kettő különbsége adja, hogy az

## Animációs effektusok

---

az animációs szakasz, amelyen az effekt hatásos, mekkora időt ölel át. Ha End a kisebb, az animáció visszafelé játszódik. Például Start=0, End=3 esetén az animáció 3 másodperc eseményeit szimulálja.

**Explosion Timing Delay %** - A robbanás késleltetése a tengelyponttól távolodva. Minél nagyobb ennek az aránya, annál lassabban terjed szét a robbanás a tárgyon.

**"Hold at" Frame Number** - A robbanás késleltetése képkockákban megadva. Csak ennyi képkocka után kezdenek a darabok szétválni.

**Bounce Particle** - Kapcsoló, melynek aktiválása után a darabok a megadott Z magasságon lévő X-Y síkra visszahullva megállnak, esetleg ha az elasztikuságuk lehetővé teszi, pattogni kezdenek.

**Flip on Bounce** - Kapcsoló, melynek aktiválása után a pattanó darabok minden pattanás során megváltoztatják forgási irányukat.

**Ground Z Position (Imagine Units)** - Annak az X-Y síknak a Z koordinátája, amelyen a tárgy darabjai a Bounce Particle opció használata esetén megpattannak, vagy megállnak.

**Use Maximal/Minimal/Average Bounding Sphere**  
- Ha ezen kapcsolók közül egyik sincs aktiválva, az elemek akkor pattannak meg a pattanás síkján, mikor tömegközéppontjuk eléri azt. Valamely itt részletezett kapcsolót kiikszelve az ütközés vizsgálatához az egyes elemek helyett azok méretétől függő gömböket vesz figyelembe a program. A Maximal hatására olyan gömb szerepel az ütközésvizsgálatban, amely az adott elem minden egyes pontját körül fogja. Minimal esetén olyan gömb lesz az ütközésvizsgálat alapja, amely az adott elem minden pontján belül helyezkedik el, míg az Average választásakor a tömegközéppont körül a pontok távolságának átlaga sugarú gömb.

## Animációs effektusok

---

**Elasticity Min/Max (%)** - Az elemek rugalmassága, energiájuk ekkora része marad meg az ütközés után.

**Initial Velocity Range Min/Max (m/s)** - A robbanás kezdeti energiája. Az egyes darabok ekkora határok közötti sebességgel kezdenek távolodni az eredeti pozíciójuktól.

**Triangle Rotations Min/Max (rot/s)** - A szétrobbanó elemek által másodpercenként végzett fordulatok minimális és maximális értéke. Az egyes darabok tényleges fordulatának száma e két határérték között véletlenszerűen választódik ki.

**Particle Trajectory Min/Max (degrees)** - A szétrobbanó elemek minimális és maximális röppályaelkelési szöge.

**Final Scaling Factor Range Min/Max** - Az elemeknek a robbanás végére elért méretük aránya az eredeti méretükhöz képest. A egyedi értékek e két határ közt alakulnak.

**Gravitational Acceleration (m/s<sup>2</sup>)** - A gravitációs gyorsulás mértéke. Ha ez negatív, a gravitáció lefelé hat, ha pozitív, felfelé. Ha értéke 0.0, nem lesz gravitáció.

**Imagine Units Per Meter** - Az Imagine egység és a méter viszonya, ennyi egység lesz egy méter.

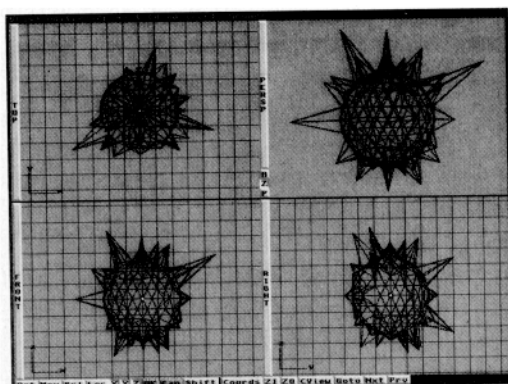
**Random Number Generator Seed** - Az effekt működését szabályzó véletlen szám, értéke bármi lehet.



## Animációs effektusok

### SPIKE

Az effektus segítségével a tárgyakra tüskéket növeszthetünk. A tárgy véletlenül kiválasztott pontjai erőteljesen kiemelkednek a tárgyból, olyan csúcsokat létrehozva



azon, mint a szöges buzogány feje, vagy egy jégkristály. A borítón lévő robbanás effektus elkészítésénél használtam, az interceptor a Shredder effekt segítségével lett darabokra szakítva, miközben a közepén egy Fire textúrával ellátott gömb volt, amelyet ez az effekt alakított egyenetlenre. Mindezt egy animáció során, a borítón lévő kép ennek egy kockája.

Paraméterek:

**Linearly, along Z** - Ha ezt a kapcsolót aktiváljuk, a tárgyból a tüskék a tárgy Z tengelyével párhuzamosan növekszenek ki.

**Cylindrically, around Z** - A tüskék másik kinövési módja, a Z tengely körül körkörösén jönnek létre, irányuk mindig merőleges Z-re.

**Radially, from Axis** - A harmadik lehetőség, a tárgyból a tüskék gömbszerűen, sugárirányban növekszenek ki.

**Minimum/Maximum Spike Distance** - A tüskék minimális és maximális hossza. Az egyes tüskék mérete e két határ között véletlenszerűen alakul. Ha ez a két érték azonos, a tüskék mindig egyforma hosszra növekszenek.

**Minimum/Maximum Spike Cycles** - Legalább és legfeljebb ennyi túske kinövési-visszahúzóási ciklus megy végbe az effekt hatása alatt.

**Dispersion Adjust (0..1)** - A tuskék által lefedett terület aránya a teljes felülethez képest.

**Random Number Seed** - A mintázat változatosságát adó véletlengenerátor kiindulási értéke.

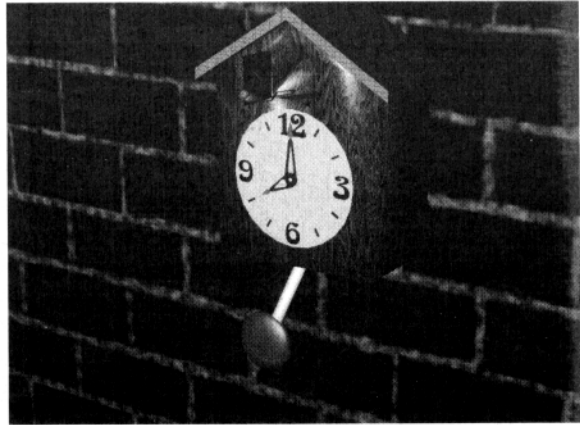
**Restrict to Integral Cycles** - Kapcsoló, melynek aktiválásával biztosíthatjuk, hogy egész számú kinövési-visszahúzóási ciklus jöjjön létre az effekt hatása során. Az animáció loopolhatóvá tételéhez hasznos.

**Reverse Spike Timing** - Kapcsoló, aktiválása az effekt fordított időbeni működését eredményezi.

**Return to Starting Shape** - Kapcsoló, melyet aktiválva az effekt végére a tárgy eredeti alakja visszaáll.

### SWAY

Az effektussal a csoportba kötött tárgyak alsóbb elemeit lehet meglóbálni. A csoport összülője mozdulatlan marad az effekt során. A lóbálás végbemehet véletlenül választott, vagy egy megadott tengely körül.



## Animációs effektusok

---

Paraméterek:

**Random/X/Y/Z Axis** - A kilengések tengelyének kiválasztását szolgáló kapcsolók. A Random hatására a csoport minden eleme véletlenül kiválasztott tengely körül leng ki. A másik három kapcsoló valamelyikével konkrét tengelyt választhatunk a lengéseknek.

**Restrict to integral gyrations** - A kapcsoló aktiválásával biztosítjuk, hogy az effekt működése során egész számú lengési ciklus megy végbe, amely elősegíti az animáció zökkenőmentes loopolását.

**Minimum/Maximum # of gyrations** - A lengések minimális és maximális száma. A csoport egyes elemei e két határ közötti számú lengést végeznek. Tört érték is lehetséges.

**Minimum/Maximum gyration angle** - A kilengések minimális és maximális szöge. A valós lengési szög minden lengés során e határértékek között véletlenül alakul.

**Random number seed** - Az effektus hatásának kialakítását végző véletlen algoritmus kiindulási értéke.

### Globális effektek

Ezen effektek közös jellemzője, hogy nem valamely tárgyhoz kell azokat rendelni, hanem a Globals-hoz. Hatásukat az egész képen fejtik ki, a kép regenerálása után hívja meg a program, hogy azt módosítsák. Némely Global Effect-nek olyan a hatása, mint valamely képfeldolgozó program egy-egy operátora. Létrehozhatunk például fadet, azaz lekeverhetjük a képet egy színre, mondjuk feketére, ami az animáció végén hatásos zárást adhat. Olyan effekt is van, amely a kész kép színeinek számát lecsökkentve és az éleket kontrasztossá téve, rajzolt kép hatását

kelti. Az effektek egy része animálható, ami annyit jelent, hogy hatásuk nem csak azonnal, hanem fokozatosan is jelentkezhethet, például az előbb említett rajzfilmszerű hatást folyamatosan veheti fel a kép, átalakulva a normál renderelt kinézetből. Szinte minden globális effektus a normál módon elkészült kép pixeleinek színét változtatja meg, de van olyan, amely a pixelek pozícióját is változtatja.

Ha olyan animációt szeretnénk, amelynek elején a képkockákon fokozatosan érvényesül egy globális effekt, majd az anim további részében a kialakult hatás állandósul, kétszer kell alkalmazni az adott effektust, egyszer animáltan, utána pedig animálatlanul.

### CEPIA (Cepiatone)

Típus:Anim, Color

Ez egy animálható Global Effect, hatására kép egy megadott árnyalatúvá változik át, például fekete színt megadva fekete-fehér, vagy barna színt megadva antik megsárgult kinézetű képeket készíthetünk. Az effektet animálhatjuk is, ekkor a megadott szín az effekt működése során fokozatosan, az effekt végére lesz teljesen domináns.

Paraméterek:

**Cepiatone Color** - Az a szín, amelyet alkalmazni kívánunk a képre.

**Reverse Timing** - Kapcsoló, melyet aktiválva az effekt fordítva megy végbe, az animáció a módosított képekkel kezdődik, majd a végére visszaáll a normál állapot.

## Animációs effektusok

---

**Return to Start** - Hatására az animáció oda-vissza játszódik le, a végére visszaáll az eredeti állapot. Ha a Reverse Timing is be volt kapcsolva, az animáció módosított képekkel kezdődik, ezek az animáció közepére átalakulnak módosíthatatlanná, majd az animáció végére visszaalakulnak módosíthatóvá.

**Don't Tween FX** - Kapcsoló, mellyel kiiktathatjuk az effekt animációs hatását. Aktiválásával az átszínező hatás már az effekt első képkockáján teljes intenzitással érvényesül.

### CONTRAST

Típus: Anim, Color

Segítségével az elkészült képek kontrasztját változtathatjuk. A változás végbemehet azonnal, vagy fokozatosan.

Paraméterek:

**Contrast Adjust (-1..1)** - A kontraszt változtatásának mértéke. Negatív érték csökkenti, pozitív növeli a kép kontrasztját.

**Intensity threshold (0..255)** - Intenzitásérték, amelyet az effekt a kulcsolásához használ fel. Pozitív Contrast Adjust esetén a kialakuló színek ettől távolodnak, negatív érték esetén ehhez az értékhez közelednek.

**Reverse Timing** - Az effektus hatását időben megfordító kapcsoló, aktiválása után az effekt a legelső kockán fejt ki teljes erővel hatását, majd az animáció végére elhal.

**Return to start** - Oda-vissza hatást létrehozó opció, aktiválva az animáció végére visszaáll az eredeti állapot.

**Don't tween FX** - Kapcsoló, melynek aktiválása után az effekt hatása azonnal, átmenet nélkül jelentkezik.

### FADBLACK (Fade to Black)

Típus: Anim, Color

Fade effektus, mellyel valamilyen színbe úsztatjuk át a képeket, például feketébe lekeverve ezzel az animáció végét. Az effekt értelemszerűen csak fokozatosan fejtheti ki hatását, ezért nincs lehetőség az azonnali hatásra.

Paraméterek:

**Fade out color R/G/B** - Az a szín, amelyre a képet átúsztatjuk az effekt hatásának végére.

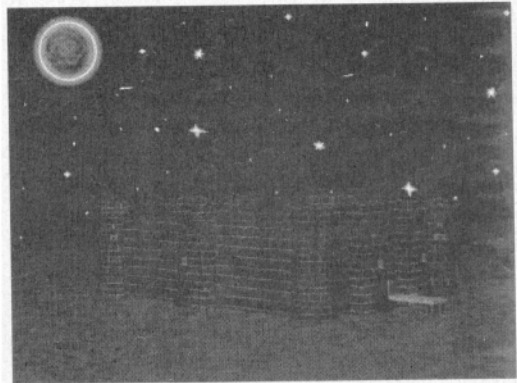
**Reverse timing** - Az effektus időbeni lefolyását megfordító kapcsoló, aktiválása után a Fade out color-ból alakulnak át a képek eredeti színükre.

**Return to start** - Oda-vissza hatást eredményező kapcsoló.

### HAZE

Típus: Color

Ennek segítségével párasságot, fényudvart, vagy hasonló hatást válthatunk ki. Az effekt egy megadott színt keres a képen és az ilyen színű pixeleket használja fel a működéséhez. Például



egy fényforrás mögé úgy rakhatunk fényudvart a segítségével, hogy a lámpa mögé egy bright attribútumú korongot helyezünk, a korong színét pedig beírjuk a Haze megfelelő kérdezőibe.

## Animációs effektusok

---

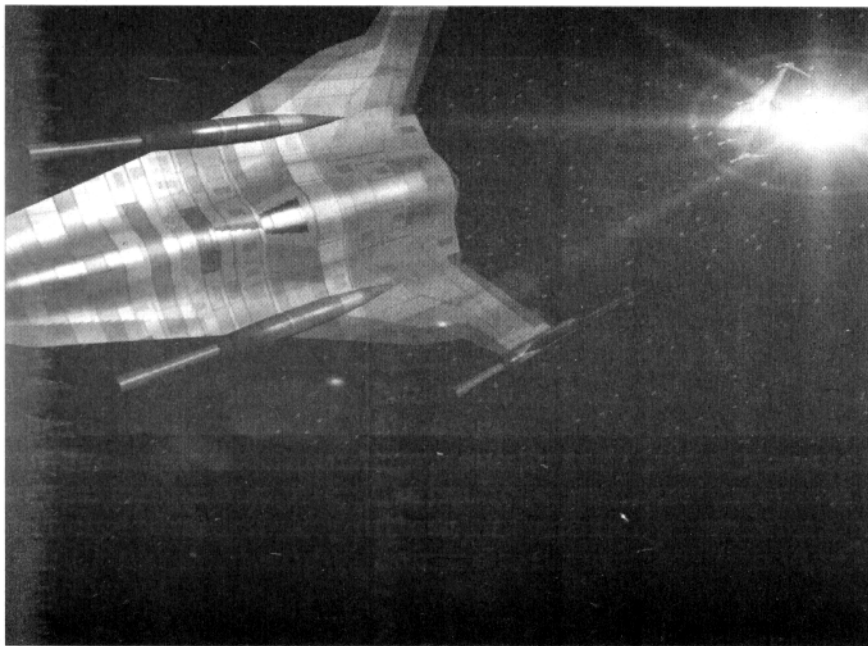
Paraméterek:

**Start/End Radius** - Az effekt nem közvetlenül a kulcsolt színű részt alakítja párássá, hanem attól egy Start Radius távolságra kezdődő és End Radius távolságra befejeződő területrészt. A távolságok pixelben vannak megadva, tehát ugyan azt a beállítást kétszer nagyobb szélességű és magasságú képen elkészítve a fényudvar fele akkora lesz. Ezt úgy lehet kiküszöbölni, hogy a nagyobb képen nagyobb sugarakat kell megadni.

**Start/End Haze Intensity** - A fényudvar intenzitása Start és End távolságra a kulcsolt színtől.

**Haze Key Color** - Az a szín, amely kiváltja a fényudvar képződését.

### LENSFLAR (Lens Flare)



Típus: Color

Ez az effektus az erős fénybe néző kamera lencséin létrejövő csillanást utánozza. A hatás csak akkor jelentkezik, ha a kamera látóterében van olyan fényforrás, amelynek "Enable Lens Flare" kapcsolója aktiválva van. A csillanás mértékét és formáját a lencsék száma, az írisz alakja és az egyes lencsék bevonata befolyásolja, ezek mindegyikét szimulálni tudjuk.

Paraméterek:

**Halo Around Light Sources** - Kapcsoló, melyet aktiválva a fényforrás körül fényudvar jön létre. Ennek méretét a fényforrás Z tengelyének méretével tudjuk szabályozni. Ez a kamera első lencséjének tökéletlenségét, vagy szennyezettségét emulálja.

**Number of Round Lenses** - A kamera kerek lencséinek száma. Ezek mindegyike egy-egy csillanást hoz létre. Előfordulhat, hogy valamely csillanás nem látható, mert kívül esik a képen.

**Number of Oktagonal Lenses** - A nyolcszögű csillanások száma. Ezeket a kamera írisze hozza létre. Előfordulhat, hogy nem látható, mert kívül esik a képen.

**Min/Max Flare Intensity (0..1)** - A csillanások intenzitásának határértékei. A maximális 1.0 érték azt eredményezi, hogy a csillanás megjelenésének helyén teljesen letakarja a képet. A 0.5 érték hatására a csillanás alatt fele intenzitással átlátszik az eredeti kép.

**Min/Max Flare Size (0..1)** - A csillanások mérethatárai, a kép szélességének arányában.

**Flare Spacing Adjustment** - A csillanások közötti távolság mértéke.

**Fraction w/o Edges (0..1)** - Egy valódi kamerában konvex és konkáv lencsék is vannak, amelyek nem egyforma csillanásokat okoznak. A konvex lencsék



## Animációs effektusok

---

intenzívebb közepű, kifelé gyengülő, míg a konkáv lencsék gyenge közepű, de intenzívebb gyűrűjű csillanásokat okoznak. Ez az érték a konvex és konkáv lencsetagok aránya.

**Random Number Seed** - Az effekt változatosságát befolyásoló véletlen szám generátor kiindulási értéke.

**Restrict to far/near side of center** - A csillanások megjelenésének helye. Előző kapcsoló aktiválása után csak a kép közepétől távolabb, míg utóbbi kapcsolót kiikszelve, csak a kép közepére eső csillanások jelennek meg.

### MELT

Típus: Anim, Color, Warp

Ezzel az effektussal a FadBlack-hoz hasonlóan kép be- és kiúsztatásokat készíthetünk. Az effekt három ilyet tud, ezek közül három kapcsolóval választhatjuk ki a megfelelőt.

Paraméterek:

**Slide/Run/Collapse** - Az effekt három módja közül az egyiket kiválasztani szolgáló kapcsolók. A slide hatására a kép alulról kezdve elhalványodik, a Run hatására ez egész kép halványodik el, míg a Collapse hatására a kép felülről leomlik.

**Horiz/Vert Frequency** - A szeletek mérete vízszintesen és függőlegesen.

**Background color** - A háttér színe azokon a helyeken, ahol az effekt már kifejtette hatását.

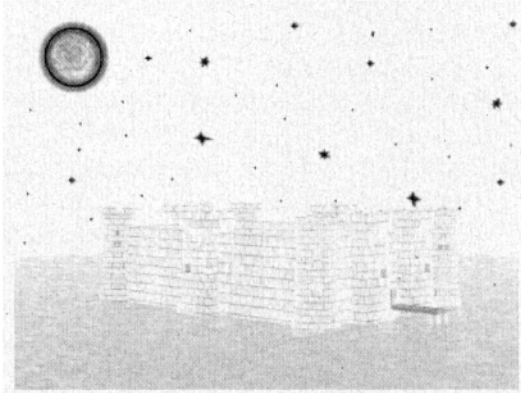
**Reverse timing** - A szokásos fordítva lejátszó kapcsoló.

**Return to start** - Ez is a szokásos, hatására az anim oda-vissza játszódik le.

### NEGATIVE

Típus: Anim, Color

Az effektus negatívba fordítja át a képet, vagy képeket. Az átfordítás fokozatosan is végbemehet.



Paraméterek:

**Reverse timing** -

Fordított hatás, a negatív képek fordulnak vissza pozitívba.

**Return to start** - Oda-vissza lejátszás.

**Don't tween FX** - A fokozatos hatás kikapcsolása.

### NEWFLARE (New Lens Flare)

Típus: Anim, Color

Hhasonló a LenFlar-hez, az erős fényforrásra tekintő kamera lencséin kialakuló csillanásokat utánozza. Két dologban különbözik azonban a régitől, egyik, hogy figyelembe veszi, ha a fényt blokkolja valamely tárgy, a másik, hogy átlátszó tárgyakon keresztül is jelentkezik a csillanás.

Paraméterek:

**Halo Around Light Sources** - Kapcsoló, melyet aktíválva a fényforrás körül fényudvar jön létre. Ennek méretét a fényforrás Z tengelyének méretével tudjuk szabályozni. Ez a kamera első lencséjének tökéletlenségét, vagy szennyezettségét emulálja.

**Number of Round Lenses** - A kamera kerek lencséinek száma. Ezek mindegyike egy-egy csillanást hoz létre. Előfordulhat, hogy valamely csillanás nem látható, mert kívül esik a képen.

## Animációs effektusok

---

**Number of Oktagonal Lenses** - A nyolcszögű csillanások száma. Ezeket a kamera írja létre. Előfordulhat, hogy nem látható, mert kívül esik a képen.

**Min/Max Flare Intensity (0..1)** - A csillanások intenzitásának határértékei. A maximális 1.0 érték azt eredményezi, hogy a csillanás megjelenésének helyén teljesen letakarja a képet. A 0.5 érték hatására a csillanás alatt fele intenzitással átlátszik az eredeti kép.

**Min/Max Flare Size (0..1)** - A csillanások mérethatárai, a kép szélességének arányában.

**Flare Spacing Adjustment** - A csillanások közötti távolság mértéke.

**Fraction w/o Edges (0..1)** - Egy valódi kamerában konvex és konkáv lencsék is vannak, amelyek nem egyforma csillanásokat okoznak. A konvex lencsék intenzívebb közepű, kifelé gyengülő, míg a konkáv lencsék gyenge közepű, de intenzívebb gyűrűjű csillanásokat okoznak. Ez az érték a konvex és konkáv lencsetagok aránya.

**Random Number Seed** - Az effekt változatosságát befolyásoló véletlen szám generátor kiindulási értéke.

**Restrict to far/near side of center** - A csillanások megjelenésének helye. Előző kapcsoló aktiválása után csak a kép közepétől távolabb, míg utóbbi kapcsolót kiikszelve, csak a kép közepére eső csillanások jelennek meg.

### RGBMIX (Red/Green/Blue Mixer)

Típus: Anim, Color

Ezzel az effektel igencsak összekavarhatjuk a kép színeit. Lekeverhetjük a képet fekete-fehérbe, vagy más színre, például kék-fehérbe, vagy akár hamis színeket alakíthatunk ki, mint egy marsbéli táj.

Paraméterek:

**newR/newG/newB** - Az új színek összetevőit kialakító egyenletek. Minden színösszetevő új értéke a három régi összetevő értékéből és egy független hozzálékből képezhető.

**Reverse timig** - A időben fordított működés kapcsolója.

**Return to start** - Oda-vissza működés.

**Don't tween FX** - Azonnali hatást kiváltó kapcsoló.

### SOLARIZE

Típus: Anim, Color

Az effekt a képfeldolgozó programokból jól ismert Solarize funkciót valósítja meg. Ez olyan látványt ad a képnek, mintha az erős nap-sütésben készült volna. Eddig ilyen nem volt lehetséges, hiába emeltük a fényforrások erejét.

Paraméterek:

**Reverse timing** - Az effekt időbeni hatásának megfordítása.

**Return to start** - Visszatérés az eredeti állapotba.



## Animációs effektusok

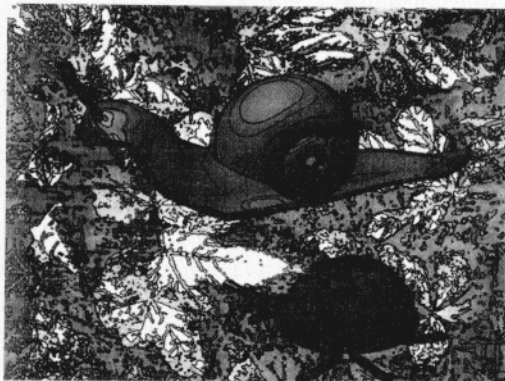
---

**Don't tween FX** - Azonnali, minden átmenetet nélkülöző hatás kapcsolója.

### TOON

Típus: Anim, Color

Ez az effekt a renderelt képekből rajzfilmszerű képeket készít. Igazán jó hatást csak kevés részletgazdag képeken eredményez, hisz az igazi rajzfilmek is ilyenek. Az



effekt fokozatosan is érvényesülhet, olyan hatást létrehozva, mintha a renderelt képek átúsznának rajzoltba.

Paraméterek:

**Draw Lines** - Kapcsoló, melyet aktiválva az effekt egy meghatározott színnel kihúzza az éleket.

**Line Color** - Az a szín, amellyel az előző kapcsoló aktiválása után az effekt kihúzza az éleket.

**Blur Lines** - Mivel az effekt nem aliasolja a kihúzott éleket, azok széle nagyon egyenetlen lesz. Ezt a kapcsolót aktiválva azonban a program szélesebbre húzza a vonalakat és elmossa a szélüket, simább rajzolatot létrehozva.

**Output Lines Only** - Ha ezt a kapcsolót aktiváljuk, az egész kép egy megadott színű lesz, amelyre a program csak a tárgyak körvonalait rajzolja fel.

**Background Color** - A háttér színe az előző opció választása esetén.

**Output Reduced Palette** - Hatására a program csak egy csökkentett színű palettát használ a kép megrajzolásához. Ez a paletta nem a kép eredeti színei alapján készül. A színek száma külön szabályozható.

**Palette Reduction Factor** - A csökkentett színű paletta színszámának faktora. A valós színmennyiség e szám harmadik hatványa lesz. Mivel a generált paletta nincs összefüggésben a kép eredeti színeivel, nem föltétlen lesz a paletta minden színe használva és hamis színek is feltűnhetnek.

**Output Full Palette** - Hatására az eredeti kép megmarad a generált élek mögött, mintegy rárajzolja az éleket a generált képre.

**Reverse Timing** - Fordított időbeni lefolyás kapcsolója, ha az effekt hatása fokozatosan érvényesül. Azonnali hatással használt effektnél nincs hatása.

**Return to Start** - Az effekt alkalmazásának végére visszatér az eredeti állapot.

**Don't Tween FX** - Kapcsoló, melyet aktiválva az effekt hatása azonnal érvényesül és nem fokozatosan.

**FX Phase (Tween Factor)** - Ha az előző kapcsolót aktiváltuk, ezzel megadhatjuk, hogy az effektus hatása mekkora mértékben érvényesüljön. Például a 0.5 érték hatására az effekt úgy befolyásolja a képet, mint a fokozatos hatás esetén az átalakulás közepén.



Újabb példa a Fireworks-ra



A tehénkéken a Jersey textúra látható

## Tippek és trükkök

A könyv utolsó fejezetében információs és tanácsdó iroda szolgáltatásai következnek, konkrét példákon keresztül adok tanácsot a képkészítéshez, ismertetek néhány trükköt és megpróbálok tippeket adni a program használatához.



### Falvastagság képezése

A primitív Tube segítségével létrehozhatunk nyitott, vagy félzárt csövet, esetleg rudat (amely nem tömör), de nem tudunk olyan olyan tárgyat kreálni, amelynek van falvastagsága. Utólag azonban könnyedén adhatunk falvastagságot a tárgynak az Extrude funkció segítségével. Nézzük hogyan. Hozz



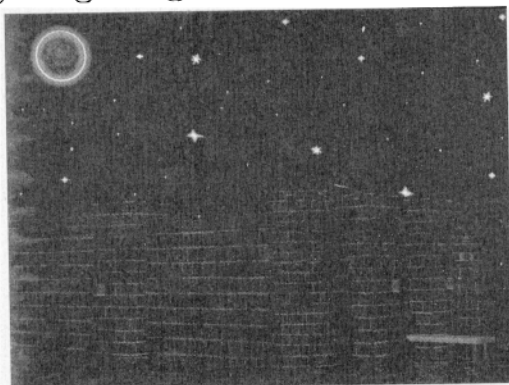
## Tippek & trükkök

---

létre egy Tube-t, majd szelektáld. Állítsd be a tengelyét, hogy hosszirányba mutasson. Most válaszd ki az Extrude-t. A tárgyat egy szegmensből kell kinyomni, de minimális távolságra. Sajnos kinyomási hosszak nem adhatunk meg nullát, pedig ebben az esetben ez lenne az optimális, de a 0.0001 gyakorlatilag majdnem zéró. Az X és Z scale értékek az elérendő falvastagságtól függenek, mondjuk legyen 0.8 mindkét esetben.

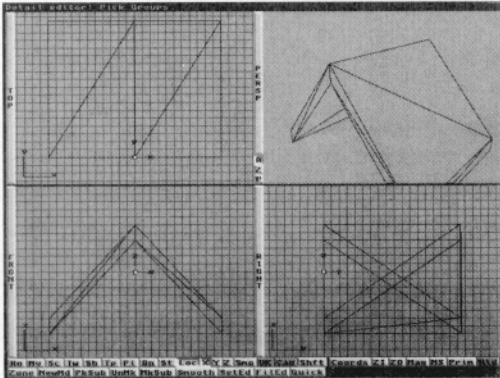
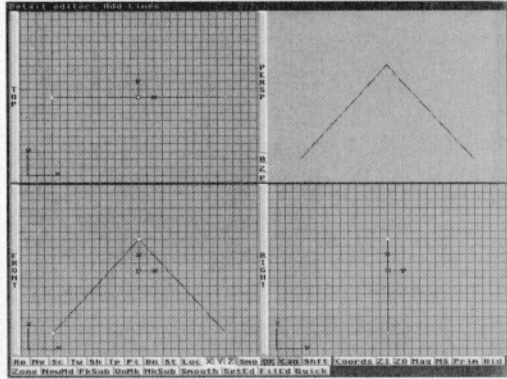
Precízebb munkák esetén még ez a piciny kinyomás is zavaró lehet, ekkor inkább nagyobb távolságra nyomjuk a tárgyat, majd Pick Pointsban kiválasztva az új pontokat, visszamozgathatjuk azokat az eredetiekkel egy magasságra.

A módszer nem csak primitív Tube, hanem bármely más, bonyolultabb felépítésű tárgy esetén is használható. A vízivárat ábrázoló képen lévő bástyák lőrésai is így készültek. Először egy csövön kialakítottam a réseket, majd ezek elkészülte



után készítettem el a falvastagságot. Mivel a tárgyak nem tömörek, a zárt végű csőből kivágva a lőrésüket, nem ugyan ezt kapjuk eredményül, ennek elérése még plusz munkát igényel. A falvastagságot azért kell utólag kialakítani, mert így könnyebb és gyorsabb a rések elkészítése.

Ugyancsak az Extrude funkciót kell segítségül hívni, ha párhuzamos, de esetleg eltolt felületeket akarunk létrehozni. Például hozzuk létre a kakukkosórán látható tetőt (lásd a 185. oldalon, a Sway effektnél). Először kell egy tengely, majd Add Lines-ben húzd meg a képen látható két vonalat. Állítsd be a tárgy tengelyét úgy, hogy az Y tengely a tető csúcsának irányába nézzen. Most jön az első Extrude, kb 40 egység hosszan. Ezután forgasd el a tengelyt, hogy az Y merőleges



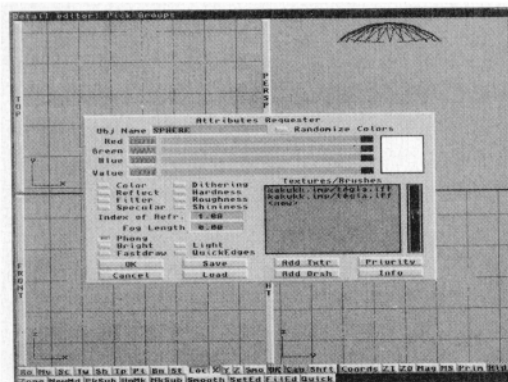
legyen az újlag kialakult felületre. Megint Extrude, most 150 egység hosszan és kész is a tető, amihez csak két élet kellett letenni három ponttal, felület készítésével pedig nem is kellett bajlódni.

### Kibontás csoport mélyéről

Bonyolult felépítésű, sok tagból álló csoportból kibontani egy elemet felvet egy problémát, nevesül, amikor a kibontandó tárgy csoportkapcsolatát megszüntetjük, a közvetlen testvérei is lebomlanak, mivel a szülőből kiinduló minden csoportkapcsolat megszűnik az Ungroup hatására. Ezen segíthetünk, ha Pick Objects módban kiválasztjuk a megfelelő tárgyat, Cut-tal kivágjuk, majd Paste-val visszamásoljuk. Így tényleg csak ez az egy tárgy válik ki a csoportból.

### Brush, textúra duplikálása

Sokszor jutunk olyan helyzetbe, hogy egy brusht, vagy textúrát meg kell duplikálni a tárgyon. Tegyük fel, ugyan azt a képet szeretnénk felhelyezni Color és Altitude mapként, de azonos paraméterekkel, hogy pontosan fedjék egymást. Addig, míg alaphelyzetben rakjuk fel a képet, nincs semmi probléma, mindkét esetben egyformán helyeződik fel a kép, de problémák jelentkeznek, ha már állítani is kell rajtuk. Legjobb volna, ha lenne egy olyan funkció, amellyel duplikálhatnánk a már felhelyezett brusht, majd a másolatban átkapcsolnánk a mapping típusát. Mivel azonban nincs ilyen funkció, marad a manuális módszer, felírni az első map paramétereit és azokat beírni a másik Transform Axes kérdezőjében. Vagy mégsem?



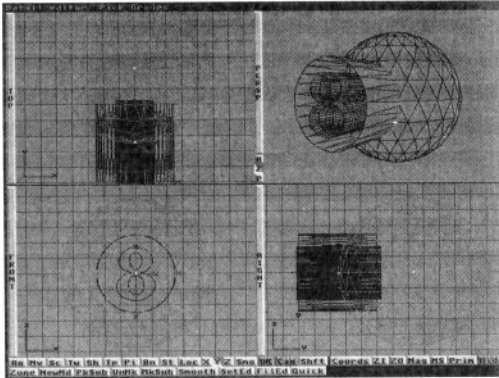
Van az Attributes kérdezőben egy piciny hiba, amelyet jól hasznosíthatunk ez esetben. Ha a Load segítségével a kérdezőbe egy külső attribútumot töltünk be, minden paraméter törlődik és felülíródik az újjal, kivéve a textúra és brush paramétereket, amelyek appendolódnak a már meglévőkhöz. Állítsd be a brusht ahogy akarod, majd mentsd ki az attribútumokat a Save-vall, ezután töltsd vissza ezt a Load-dal, máris van két azonos brush beállítás, az egyikben kapcsold át a mapping típusát és kész. Ez a hiba nem jelentkezik a Quick Attributesek használatakor, ott rendesen törlődnek a régi textúra és brush paraméterek.

### Mapping helyett alakzat

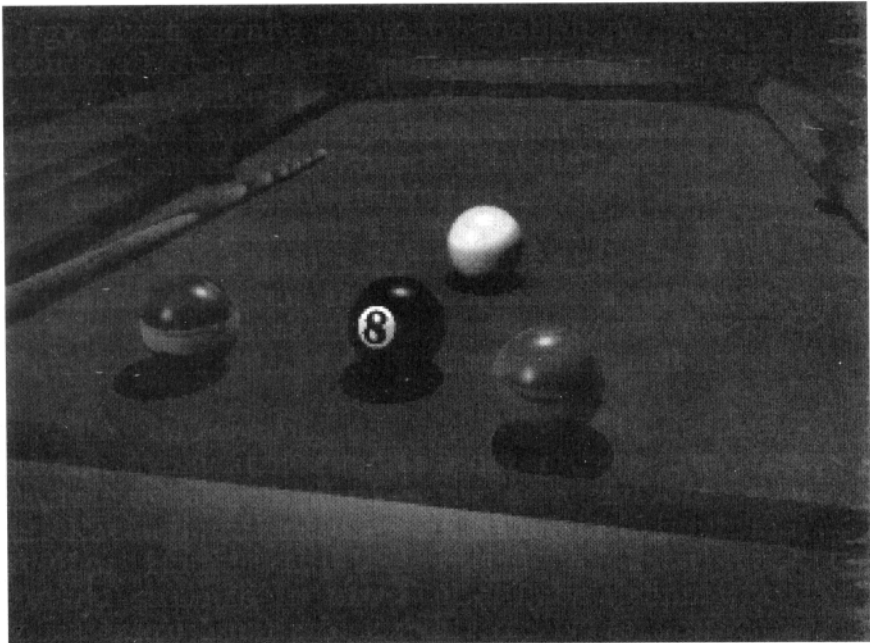
Sajnos a brush felfeszítése során nem adhatunk meg egy olyan színt, amely a brushon átlátszó, így minden esetben a teljes téglalap alakú kép felkerül a tárgyra. Ez olyankor jelent problémát, amikor kerek, vagy ovális brusht kellene felfeszíteni, mint például a biliárdgolyón a szám és a háttere. El lehet ugyan kísérletezni, hogy a brush háttérszínét azonosra állítsuk a tárgy színével, de így is meg fog látszani a kép határa. Ilyen esetekben inkább készítsük el a megjelenítendő képet, jelen esetben a számot és a kerek hátterét objectként és fűzzük egybe a tárggyal, majd a megfelelő felületek attribútumait külön állítsuk be.

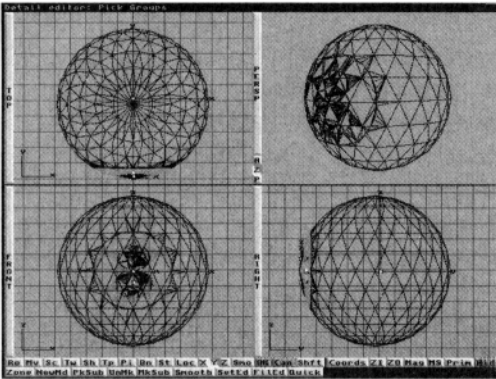
Lépj át a Spline editorba és egy ide illő betűtípusból készítsd el a nyolcast. Elegendő most a körvonal, ezért az Add Points funkcióval sem kinyoni, sem felületekkel ellátni nem kell. Mentsd ki a pontokat, majd lépj át a Detail-ba. Töltsd be a nyolcast, extrudált, majd helyezd el egy Tube primitívben, hogy közepére kerüljön. Ez lesz a kivágó szerszám,

## Tippek & trükkök



Joinold egybe azokat. Készíts egy primitív gömböt, a kivágó eszköznek állítsd be a méretét és a pozícióját, hogy a kívánt alakot hasítsa ki a gömbből. Ezután jöhet a Slice. A szeletelés végeztével keresd meg a gömb palástjának darabjait és egyenként kiválasztva azokat Pick Faces-ben a Select All segítségével a teljes felületükből hozd létre a "gömb", "kör" és "szám" alcsoportokat. A szám háttérében lévő kör három darabból áll, hozzá



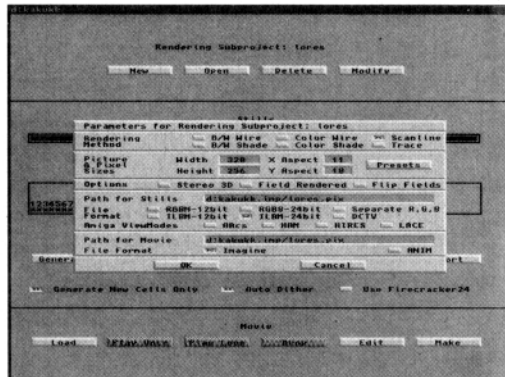


tartozik a nyolcasban lévő két bezárt terület is, ezeket Joinold egybe mielőtt átlépsz velük Pick Facesbe. A Merge segítségével egyesítsd a három gömbpalást darabot, a felesleges darabokat töröld le ezzel kész is a biliárdgolyó.

Azért kellett az egyes elemek teljes felületéből előre alcsoportokat képezni, mert így könnyebb volt kiválasztani azokat és az alcsoportbeállítások megmaradnak a Merge és a Join után is. Válaszd ki a gömböt és állítsd be a globális attribútumait, majd Pick Facesbe átlépve változtasd meg a számnak és a háttérének a színét külön külön.

## Field rendering

Van a Project editorban egy, illetve kettő, de szorosan összefüggő kapcsoló, amelyről mindeddig nem esett szó. Ez a **Field Rendering** és a **Flip Field**. Használatuk elsősorban akkor indokolt, ha az animációt képkockaként rögzítjük videóra. A Field renderinget bekapcsolva a



program a félképeket külön számolja ki, majd a két

## Tippek & trükkök

---

félképet elkészültük után egybefűzi. Például egy 768x576-os képméretű animáció esetén készít két 768x288 pixel méretű képet, amelyet úgy egyesít, hogy az egyik a páros, a másik a páratlan sorokat képezi. Mi ennek a jelentősége, ha a végén úgy is egy 768x576-os kép készül? Az, hogy a tárgyak elmozdulásait a félképek között is figyelembe veszi a rendering során, ezáltal a gyorsan mozgó tárgyakra sokkal finomabb, kevésbé vibráló képet kapunk, a mozgás lágyabb lesz. Így gyakorlatilag nem 25 kép/s, hanem 50 kép/s sebességű animációt kapunk alig hosszabb rendering árán. Az elkészült képek kezelése a továbbiakban ugyan úgy történik, mind a többié.

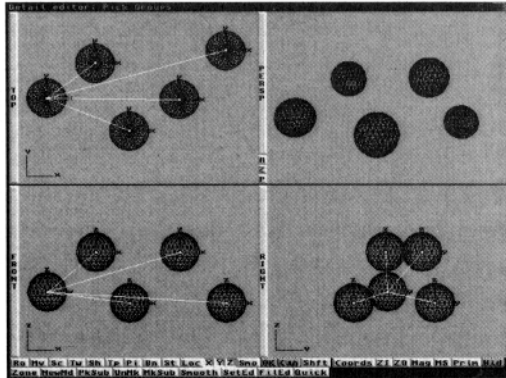
Előfordulhat, hogy a Field renderinggel készült képek a rögzítés után még jobban vibrálnak, mint a normál képek, ez azért van, mert a félképek sorrendje a rögzítőeszközön más mint a számítógépen. Például ha a gép úgy számolta ki, hogy az első kép lesz a páratlan, a második kép a páros félkép, de a videó pont fordítva rögzíti, akkor jön létre az előbb említett hiba. Ezt is kiküszöbölhetjük azonban a Flip field kapcsoló használatával, ekkor megfordul a félképsorrend, az elsőnek generált kép lesz a páros, a másodiknak generált a páratlan félkép.

Hogy melyik beállítás a megfelelő az próbálkozással kell kideríteni, mert mindig az adott felvevő eszköztől függ. Készíts egy rövid 30-40 kockás képsorozatot, amelyen csak annyi történik, hogy egy tárgy nagy sebességgel átszáguld a képernyőn. Rendereld le mindkét félképsorrenddel és ezek rögzítése során megkapod, melyik beállítás a megfelelő arra a videóra.

## Pattogó tárgyak

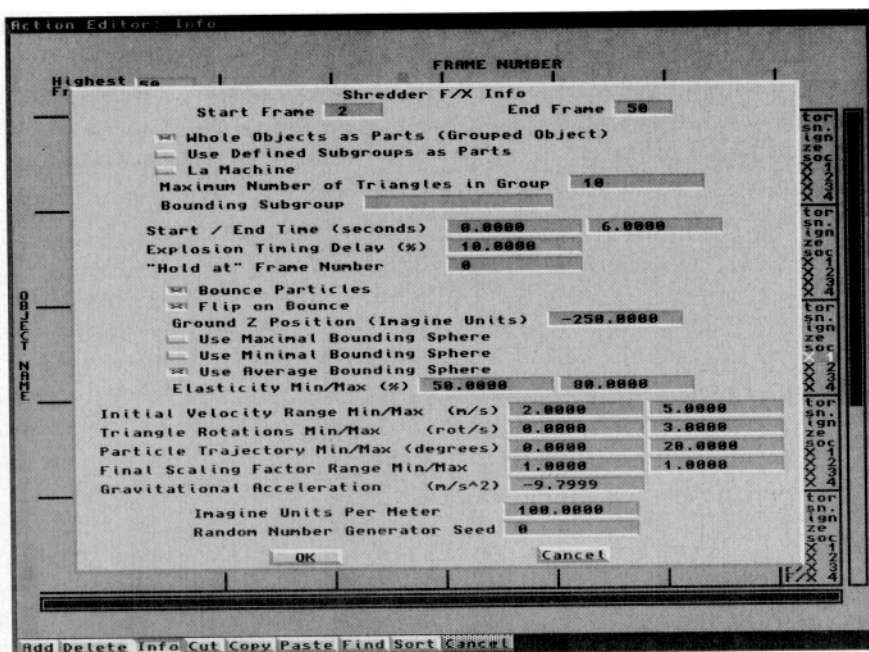
Az Imagine nem rendelkezik ütközésvizsgálattal és kinematikai modellezéssel, így valós fizikai jelenségeket csak nehézkesen, a mozgásfolyamatokat manuálisan beállítva tudunk ilyen animációkat készíteni. A Shredder effektust felhasználhatjuk azonban pattogó tárgyak készítéséhez. Ehhez az effekt azon tulajdonságai használjuk fel, hogy a szétszabdálás során a csoport egyes elemei egyben maradhatnak és hogy a lehulló darabok egy megadott Z magasságon lévő képzeletbeli X-Y síkon megpattanhatnak.

Készíts a Detail editorban öt egyforma gömböt, majd lásd el ezeket különböző textúrákkal. Kösd csoportba a gömböket, ahogy az a képen látszik. Készíts egy talajt is, tetszőleges mintázattal. Mentsd ki a tárgyakat, majd lépj át az Action editorba. Állítsd be az animáció hosszát 50 kockára, majd töltsd be a talajt és a golyókat. A talajt tedd  $Z=-250$  magasságra, a golyókat  $Z=0$ -ra. A golyókhoz adj egy Shredder effektust és állítsd be a következő paramétereiket.



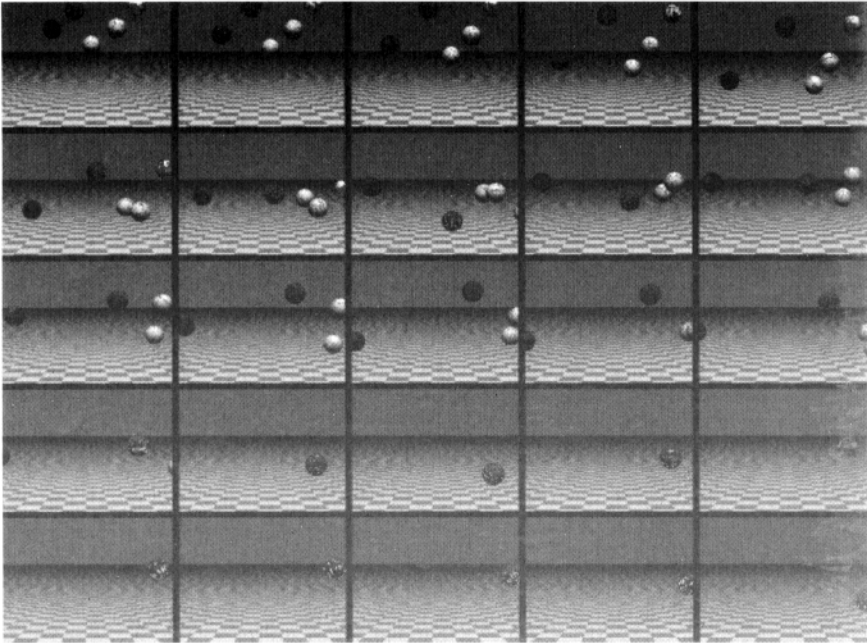


# Tippek & trükkök



Mentsd ki a változtatásokat, lépj át a Stage editorba. Állítsd be a kamerát és helyezz el fényforrást saját belátásod szerint. Preview generálásával ellenőrizd, hogy a gömbök jól pattannak-e. Ha igen, rendereld le az animacit, mert a gömbökön lévő textúrák forgása csak ekkor látszik. Jó mi?

Sajnos a tárgyak egymással való ütközését nem tudjuk vizsgálni, megeshet hogy két golyó áthalad egymáson.



## Neonfény

Az Imagine fényforrásai mind pontszerűek, vagyis a fény mindig egy pontból, a fényforrás tengelypontjából árad. Tegyük fel, neoncsövet szeretnénk létrehozni, hogyan lehet ezt megvalósítani ilyen korlátok mellett? A Sftstrip textúra kiválóan alkalmas a probléma megoldására. Ezzel mint tudjuk, három lágy szélű csíkot rajzolhatunk a tárgy X-Y síkjára az X tengellyel párhuzamosan. A csíkok a textúra tengelyeivel határolt területen belül jelennek meg.

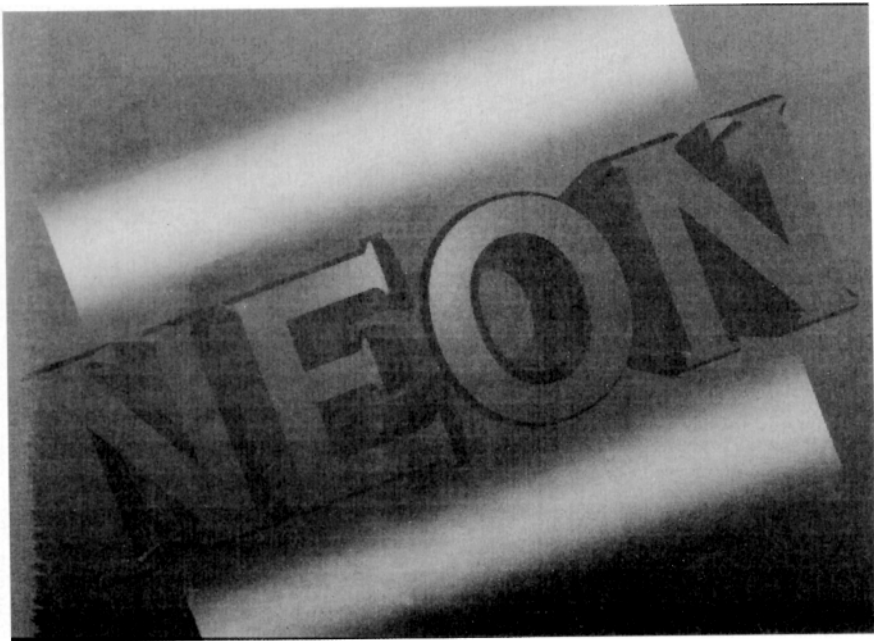
Készíts egy sík lapot 1x1 szegmensből, forgasd el a tengelyét X körül  $90^\circ$ -al, hogy Z merőleges legyen a felületre. Lépj be az Attributes-be, kapcsold ba a Bright-ot, a Filter értékeket pedig állítsd 255-re,

## Tippek & trükkök

---

hogy a tárgy teljesen átlátszó legyen. Töltsd be a Sftstrip textúrát. A Stripe 1 és 3 Width értékeinek adj 0.0-t, ezzel elérjük, hogy csak egy, a középső csík jelenjen meg. Legyen a Stripe Filter 0, azaz a csík nem lesz átlátszó. A neon színét a Stripe 2 R, G, B input mezőkben lehet megadni.

Ha kész a beállítás, az Attributes-ben kapcsold be a Light-ot, a tárgyból fényforrást csinálunk. A fény színe legyen azonos, esetleg erősebb intenzitású, mint a neon színe. Mivel a neoncső fénye nem irányított, nem kell bekapcsolni sem a Round, sem a Rectangular Shape kapcsolókat, de a Diminish Intensity-t belátásod szerint használhatod. Mivel a fény valójában egy pontból fog áradni, nem a neon teljes felületéből, a LensFlar nem fog jól kinézni, ha a képen szándékozod használni a nevezett effektet, inkább kapcsold be a No Lens Flare-t.



### Színes fényjáték

Ha már a fényeknél tartunk, tegyük fel készíteni akarunk egy animációt, amelyben a színpadot színes kavargó fény világítja meg, mint mondjuk egy didzsi táncparketjét. Erre a célra használhatunk több különböző színű fényforrást, de ezek külön-külön kezelése nehézkes, időtrabló, nem beszélve a több fényforrás okozta többlet rendering időről. Sokkal jobb hatást érhetünk el a Transpar textúra segítségével, amely a tárgyon lévő színeket átlátszóság értékkel alakítja, úgy, hogy a tárgyon áthaladó Cast Shadow tulajdonságú fényforrások fénye trace módban felveszi a megfelelő szintet. A Clrnoiz textúrával kombinálva például a fényben a színek örült tobzódását hozhatjuk létre. Készíts egy gömböt, helyezd rá először a Clrnoiz-t, majd a Transpart. Tedd a tárgyat fényforrássá és kapcsold be a Cast Shadow-t. Mivel a fény a tengelypontból ered, amely jelen esetben a gömb közepén van, a fénynek át kell haladni a gömb palástján, ahol felveszi a Clrnoiz által kialakított színeket, a fényforrás színes fényt fog sugározni (persze csak trace módban, hisz ekkor van valódi fénysugárkövetés). Ha ezt a gömböt az animáció során két, vagy három tengely körül különböző sebességgel forgatod létrejön egy csodás színes kavalkád.

A Transpart nem kell használni ha a színek kialakításához a Mosaic textúrát alkalmazod, mert az önmaga képes a fenti hatást modellezni.

### Tűzlövedék

Hogyan lehet tűzlövedéket, vagy hasonló tárgyat létrehozni, olyat, mint a borítón az X-Wing lövedékei? Természetesen ezt is az Imagine egy textúrájával. Először is készítsük el a lövedék alakját. A kiindulási tárgy a gömb. Mozgasd el a tengelyét a

## Tippek & trükkök

---

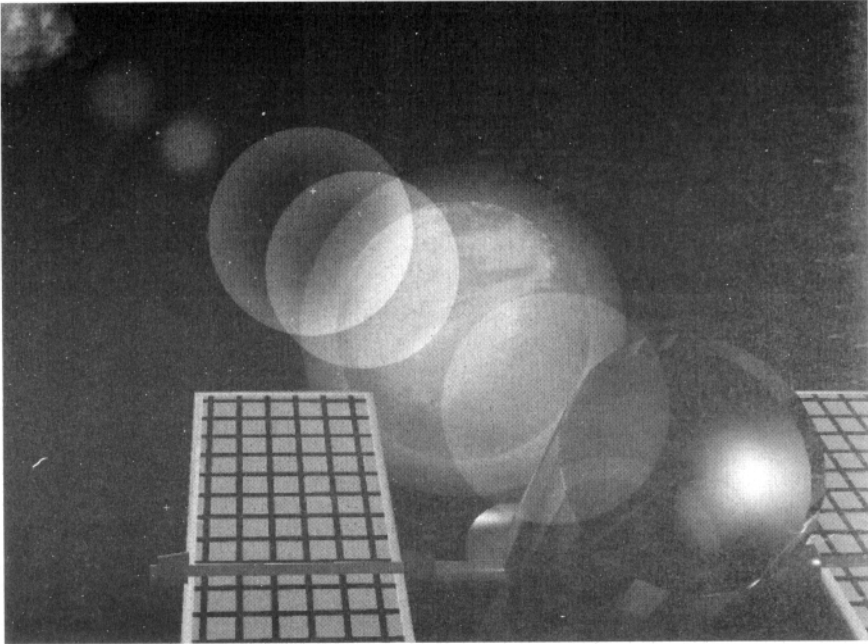
frontnézetben az aljára, majd növeld meg a Z tengelyt, hogy kissé magasabb legyen mint a tárgy. Ezzel előkészítettük a Taper torzításhoz, amely interaktív használat esetén a tengelyponttól a +Z irányban terjedő tárgyrészt torzítja. Bökj a Tp feliratú kapcsolóra és húzd össze a gömb tetejét, hogy cseppformát kapj. Ez még túl zömök, hogy lövedék legyen, nyújtsd meg a tárgyat a Z tengelye mentén. Ha kész, alkalmazzuk rá a tüzet. Ezt megtehetjük a Fire, vagy a Fireball textúrával, most az utóbbit ismertetem. Lépj be az Attributes-be és tedd a tárgyra a Fireball-t. Allítsd úgy be a textúra tengelyét, hogy a tengelypont a lövedék tompább végére essen, a tengelyhosszokat pedig, hogy kissé túlérjenek a tárgyon. A textúra kérdezőjében állítsd be a tűz attribútumait, a Noise-hoz pedig írd 0.5-öt. Ha a Fire textúrát használod, a beállítások hasonlóak, de az animálható is, a tűz lobogni fog az animáció során.

### Bolygók

Az űrbeli jelenetek, űreszközök és űrjárművek mindig is a számítógépanimációk legnagyobb ihletőinek számítottak, nézzünk néhány ezzel kapcsolatos dolgot.

Az egyik leglátványosabb bolygó maga a föld. A légköre miatt kékes aura veszi körül az űrből nézve. A látvány modellezéséhez két azonos középpontú gömb kell, az egyik 5-10%-al nagyobb legyen a másiknál. A belső gömb a bolygó, erre kell Wrap-Wrap mappinggal felfeszíteni egy föld térképet, a másik lesz a lékör. Utóbbinak legyen világoskék a színe, a Fog Length-je pedig alap méretű Sphere primitívet feltételezve 80-100 egység. A tárgynak még túl élesek a körvonalai, ezért érdemes rá alkalmazni egy Ghost textúrát. Ennek Fog Length at T értéke,

szintén csak alap méretű gömböt feltételezve 150-200 körüli legyen. A Reflect-et és a Specular-t nullára kell állítani, mivel valószerűtlen látványt nyújtana, ha a légkör csillogna, vagy tükrözne. A filter érték közömbös, mivel Fog objecten nincs hatása.



A másik szép bolygó a Szaturnusz, a maga gyűrűivel. Ennek utánzására külön textúra van, a Gasgiant. A textúra Z körül elhelyezkedő koncentrikus sávokat alakít ki a tárgyon, amelyeket mind a bolygó, mind a gyűrűinek elkészítéséhez megfelelő. Lássuk hogyan.

Hozz létre egy gömböt, ez lesz a gázbolygó, alkalmazd rá a Gasgiant-ot alap paraméterekkel.

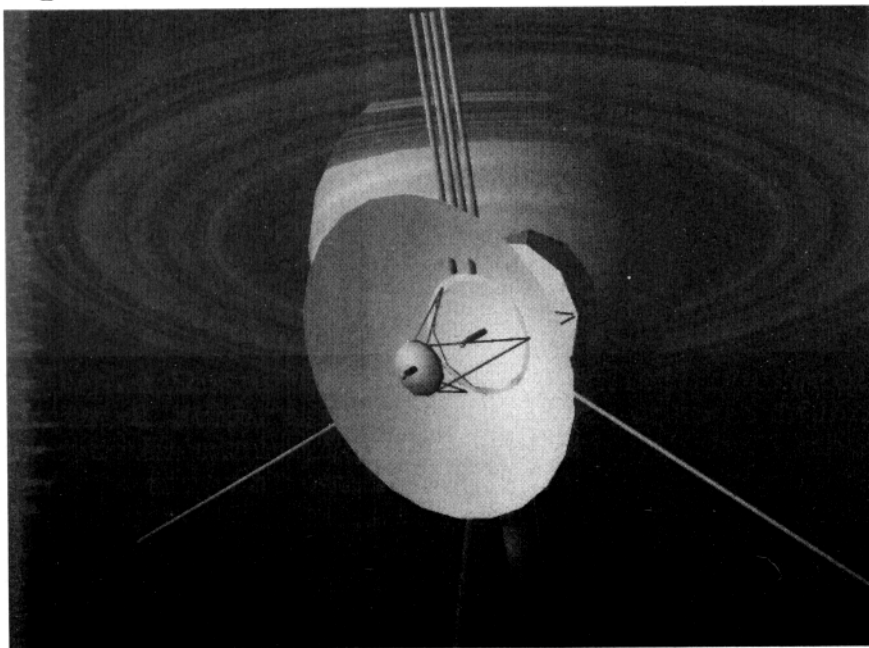
Készíts egy Tube primitívet, méretezd akkorára, hogy az átmérője 2-3 szorosa legyen a gömbnek. Adj

## Tippek & trükkök

---

neki falvastagságot a korábban ismertetett módszerrel, úgy, hogy a cső belseje és a gömb között legyen kb. egynegyed bolygó átmérőjű hézag. Ha elsőre nem sikerül eltalálni a megfelelő arányt, ne kísérletezz, inkább Pick Points módban válaszd ki a belső pontsorát a gyűrűnek és Scale segítségével állítsd be a megfelelő méretet. Ekkor azonban kapcsold ki a Z tengely menti méretváltozást, hogy a gyűrű magassága ne változzon. Apropos magasság, ha idáig eljutottunk, a csövet alacsonyabbra kell venni, hogy megfeleljen a bolygó gyűrűinek.

A gyűrű legyen teljesen átlátszó, majd alkalmazd erre is a Gasgiant textúrát. Ne felejtse el előtte a tengelyt visszaállítani, hogy a Z merőleges legyen a végére!

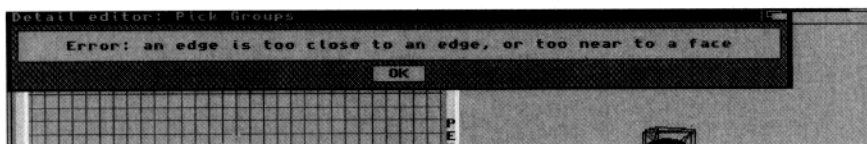


Növeld meg a Minimum Ring Radiust 50-60, a Maximum Ring Radiust pedig 150 körülire, a Filter Adjus-ot pedig állítsd 1-re. Utóbbi hatására a sávok nem lesznek átlátszóak. A gyűrűk színeit vedd sötétebbre.

Groupold egybe a bolygót és a gyűrűket, hogy könnyebben kezelhetők legyenek. Ha a renderelés során azt tapasztalod, hogy a gyűrűk száma túl sok, vagy ellenkezőleg, kevés, a Banding Width Adjust-tal tudsz ezen állítani.

### A Slice hibái

A Detail editorban amikor több tárgyat próbálunk egymásból kivágni a Slice segítségével, gyakran találkozunk az "Error: an edge is too close to an edge, or too near to a face" hibaüzenettel.



A hibát két dolog okozhatja. Az egyik, hogy két tárgy olyan pozícióban van, hogy bizonyos élei épp fedik egymás, ezért a program nem tud vágási felületet meghatározni. Ezen tudunk segíteni, ha kissé megmozdítjuk, vagy elforgatjuk az egyik tagot. Ez a mozdulás elegendő ha néhány század egység messzire történik.

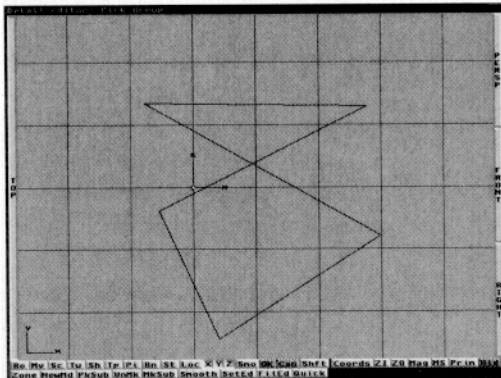
Ha több ilyen próbálkozás sem segít, akkor lehet gyanakodni, hogy máshol van a hiba. Ellenőrizd le a tárgyakat az Object/Check obj. funkcióval. Ha olyan hibaüzenet jelenik meg, hogy "Error: an edge use same points twice", akkor az azt jelenti, hogy van a tárgyban olyan pont, amely egymaga egy szakasz mindkét vége. Így az él nulla hosszú, ami megza-



## Tippek & trükkök

varja a programot. Eltávolítani úgy lehet az ilyen pontokat, hogy kiválasztod a tárgyat egymagában, majd használod az Object/Merge-öt.

Ha a Check obj. sem jelez hibát, akkor még lehet egy ok a Slice nem működésére, az, hogy valamely tárgyban önmagában van keresztező él, például olyan, mint itt a képen. Erről úgy győződhetsz meg, ha a tárgyat egyedül ki-

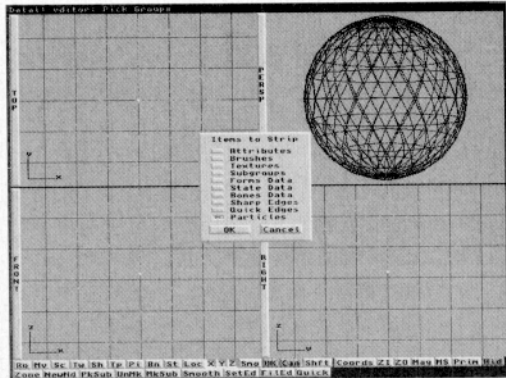


választva használod a Slice-t. Ha ekkor is megjelenik a hibaüzenet, ez az eset áll fenn. Helyes, átgondolt tárgyszerkesztés esetén ritkán jönnek létre ilyenek, sokkal gyakoribb, ha más objectformátumról konvertáltuk a tárgyat. Ismerek néhány konverter programot, amely minden lelkiismeretfurdalás nélkül hoz létre ilyen élkeresztezéseket. A másik gyakori eset a létrejöttére, amikor a Spline editorban, rosszul megszerkesztett fontot használunk.

## Stirpe

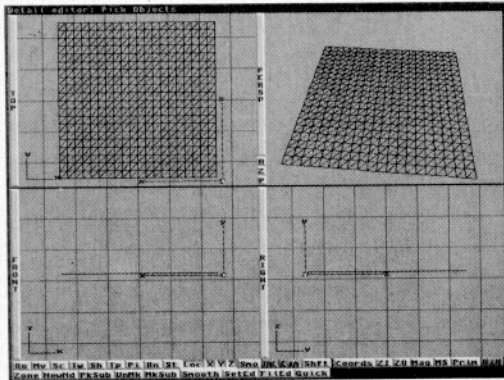
Sokszor előfordul, hogy egy tárgyat sikerül ellátni mindenféle felesleges beállítással, például rengeteg al csoportdefinícióval, statekkel, textúrákkal, stb. Ezek mind jelentősen növelik a tárgy méretét, hasznos volna megszabadulni tőlük, mondjuk törölni kellene az összes al csoportdefiníciót. Erre való az Unmake Subgroup funkció, de három tucat al csoport esetén már unalmas, hogy egyenként kell törölni azokat. Szerencsére a program kínál ehhez

egy jóval gyorsabb megoldást, amellyel egyszerre eldobhatjuk a tárgy valamely beállításait. Válaszd ki a 'levetkőztetni' szánt tárgyat, majd hívd meg a **States** menü **Strip Object** pontját. Megjelenik egy kérdező, ahol azokat a paramétercsoportokat választhatjuk ki, amelyeket törölni szeretnénk a tárgyból. Egyszerre több csoportot is kiválaszthatunk, ezek mindegyike törlődni fog. Bánj óvatosan a törölgetésekkel, mert könnyen áldozatul eshet olyan tulajdonság is, amelyre még szükség van!



## Hegyek, völgyek

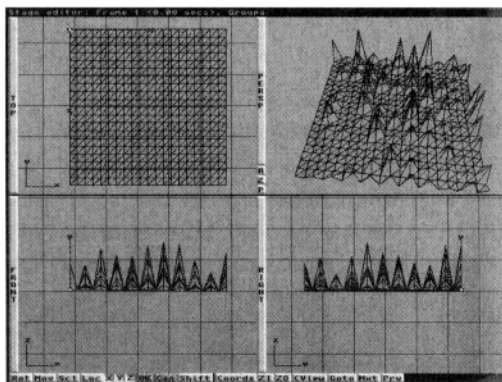
Hogyan fognál hozzá hegyet ábrázoló kép elkészítésének? Természetesen a kiindulási alap egy jó sok pontból álló síklap, amelynek pontjait függőlegesen kihúzogatójuk, akár a Magnetism segítségével, akár anélkül. Az eredmény igen szép is lehet, ha az embernek van elég türelme a sok pont közül a megfelelőket kiválasztani és elmozgatni, de véletlen alakú hegység elkészítéséhez rövi-

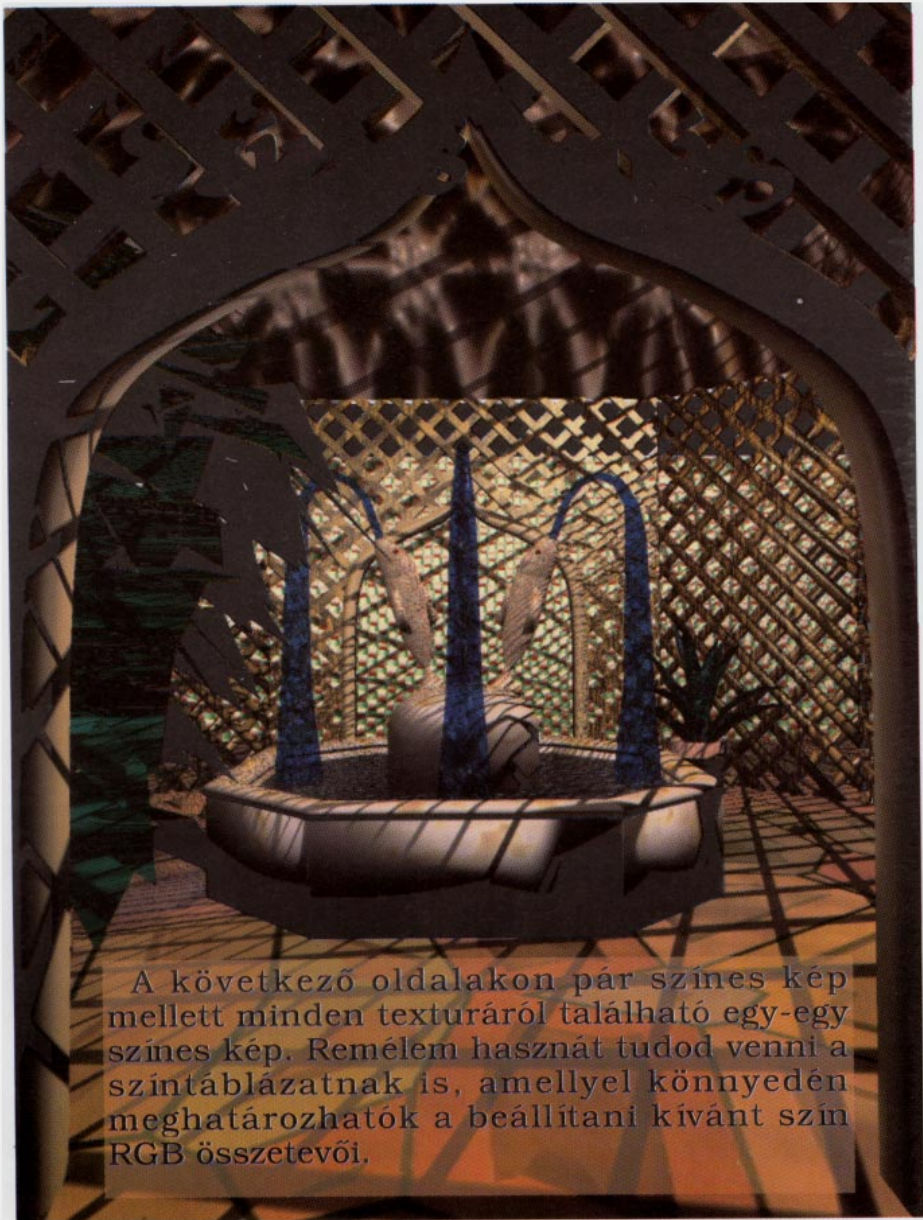


## Tippek & trükkök

debb, kényelmesebb út is vezet. Készítsd el a síkot 20x20 szeletből, forgasd el X körül 90°-al, hogy vízszintesen álljon, állítsd be a tengelyét úgy, hogy a Z merőleges legyen a síkra, a tengelypont valamely sarokba essen X és Z pedig a sík oldalain legyen, majd mentsd ki. Lépj át az Action editorba, készíts egy 10 kockás animációt, amelybe töltsd be az előbb elkészített síkot. A tárgyhoz tegyél egy Spike effektet, amelyben válaszd a Linearly, along Z kapcsolót, a tűskék hosszát állítsd be a hegy magasságának megfelelően. Lépj át a Stage editorba és keresd meg a tíz kocka közül azt, amelyen leginkább tetszik az alakzat, majd mentsd ki az Object/Snapshot segítségével. Ezt az alakzatot töltsd be a Detail editorba.

A hegy mintázatának elkészítéséhez használd a Mntntop textúrát. Persze ezek igencsak cukorsüveg-hegyek, de ha a háttérbe kellene, akkor jók.





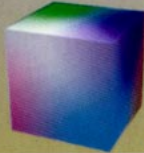
A következő oldalakon pár színes kép mellett minden texturáról található egy-egy színes kép. Remélem hasznát tudod venni a színtáblázatnak is, amellyel könnyedén meghatározhatók a beállítani kívánt szín RGB összetevői.

## Textúra áttekintés

A következő színes képek a program összes textúráját ábrázolják, beleértve az előző kötetben leírtakat is.



Agate



Angular



Brick



Brushed



Bumpnoiz



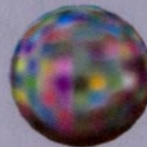
Camo



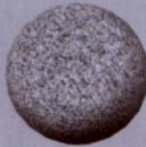
Checks



Checks2



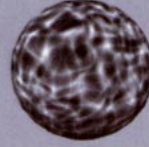
Colornoiz



Concrete



Confetti



Crumpled



Dinoskin



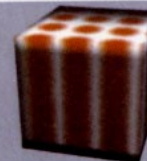
Dirt



Disturbed



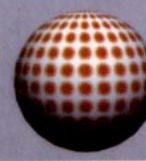
Dots



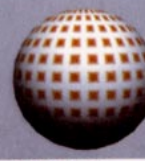
Dithcirc



Dithrect



Rdditcir



Rdditrec

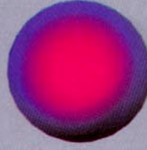
# Textúra áttekintés



**Dirtpaint**



**Easywood**



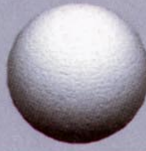
**Fakely**



**Frogskin**



**Jersey**



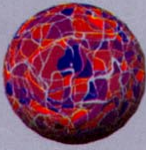
**Leather**



**Marble**



**Monster**



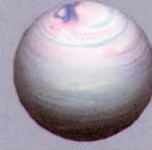
**Mosaic**



**Clrnoiz2**



**Refnoiz2**



**Filnoiz2**



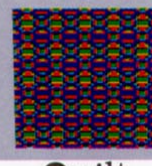
**Oldbrick**



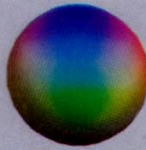
**Pebbled**



**Peened**



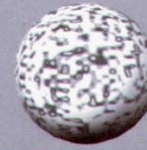
**Quilt**



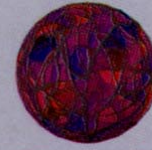
**Rainbow**



**Scratch**



**Spotch**



**Staings**



**Statue**



**Terra**



**Wormvein**

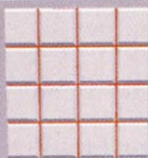


**Wrinkle**

# Textúra áttekintés



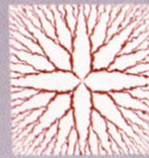
Zoolo



Battile



Blast



Bmpbrnch



Branches



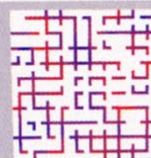
Shingles



Stamped



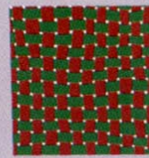
Trichex



Tracer



Tritile



Weave



Rectchex



Radcheks



Sprlchex



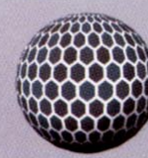
Dethstar



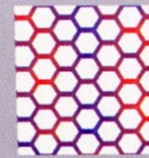
Rddthstar



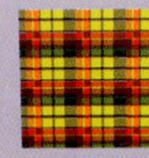
Honycomb



Radcomb



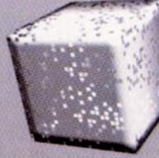
Hexez



Plaid



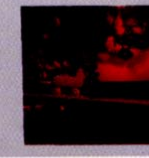
Radwinds



Rectwinds



Tubewind



Antique

# Textúra áttekintés



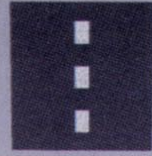
**Cndappl**



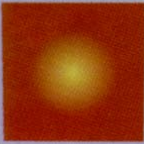
**Coolfire**



**Craks**



**Dashline**



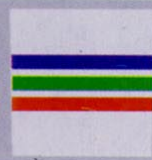
**Fireball**



**Gasgiant**



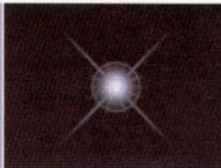
**Hardwood**



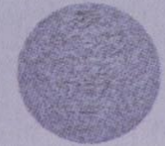
**Hrdstrip**



**Iris**



**Lensflar**



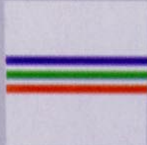
**Metals**



**Mntntop**



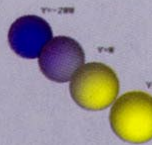
**Ribbed**



**Sftstrip**



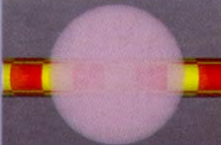
**Transpar**



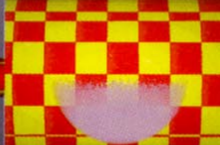
**Zbuffer**



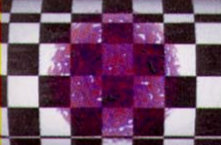
**Venitian**



**Ghost**



**Fogtop**



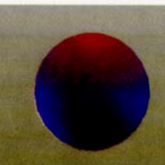
**Fogpaint**



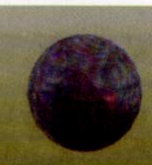
**Nebula**



**Grid**



**Linear**



**Pastella**

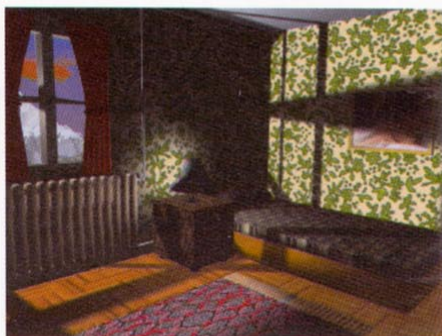


# Textúra áttekintés



Radial

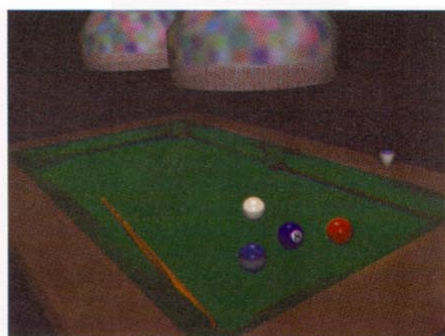
Spots



Frnchwin



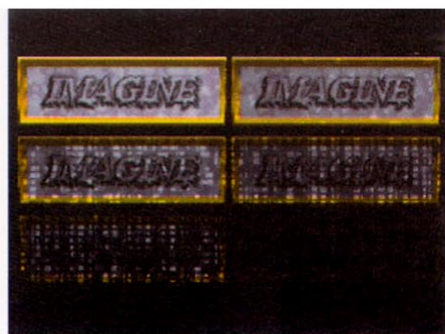
Wood



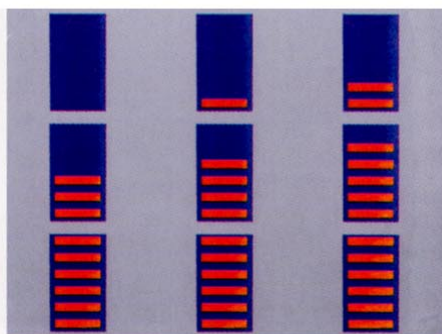
Softedge



Strobe

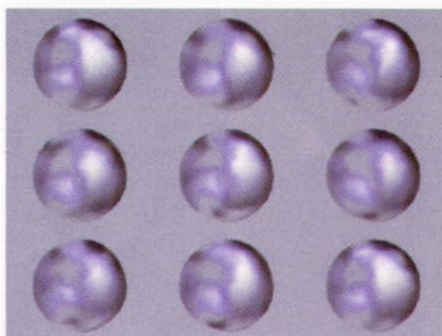


Beammeup

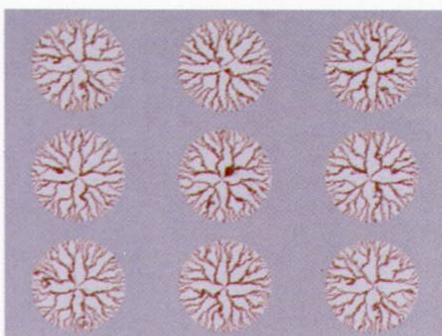


Chaser

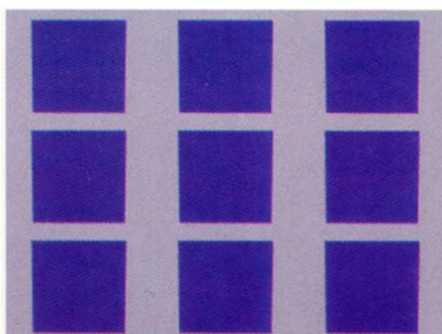
## Textúra áttekintés



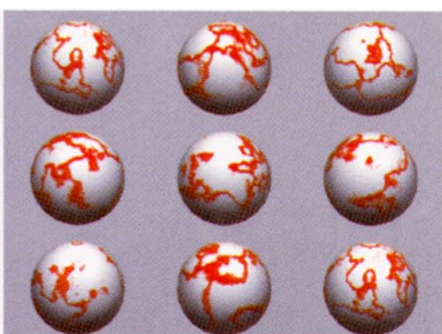
Clouds



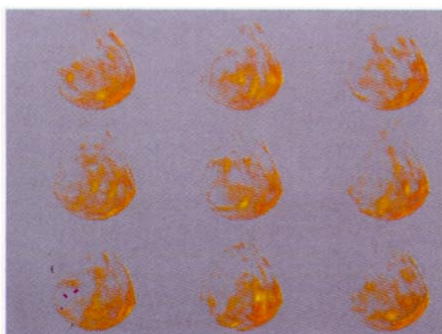
Dancspark



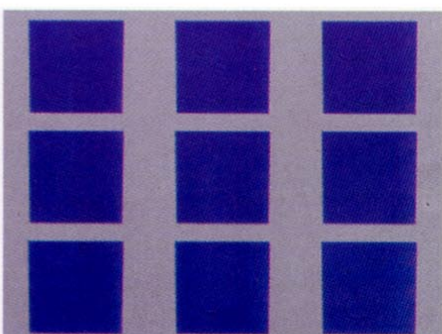
Dripdrop



Electric

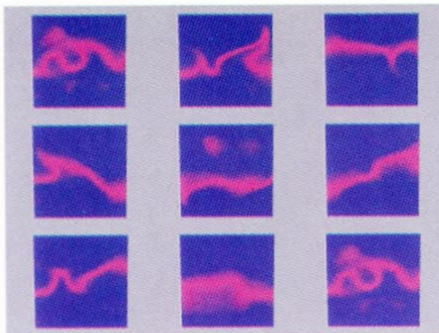


Fire

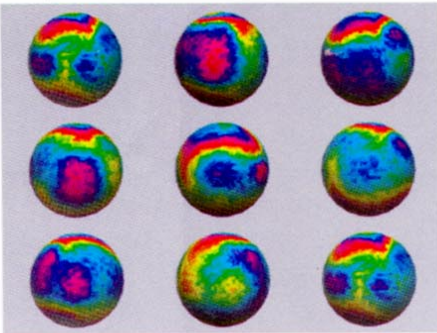


Rain

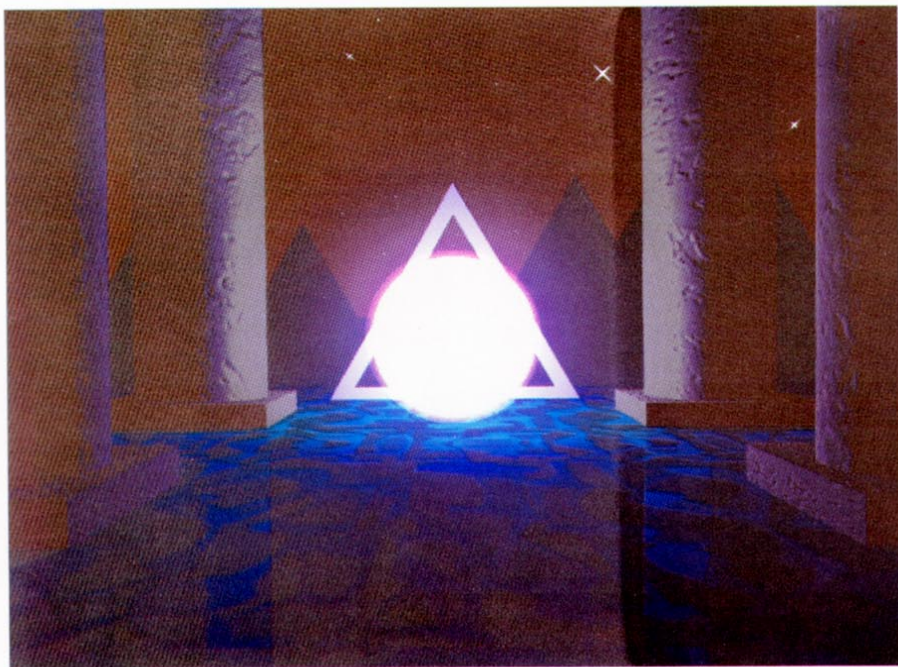
## Textúra áttekintés



Sparks



Tiedye



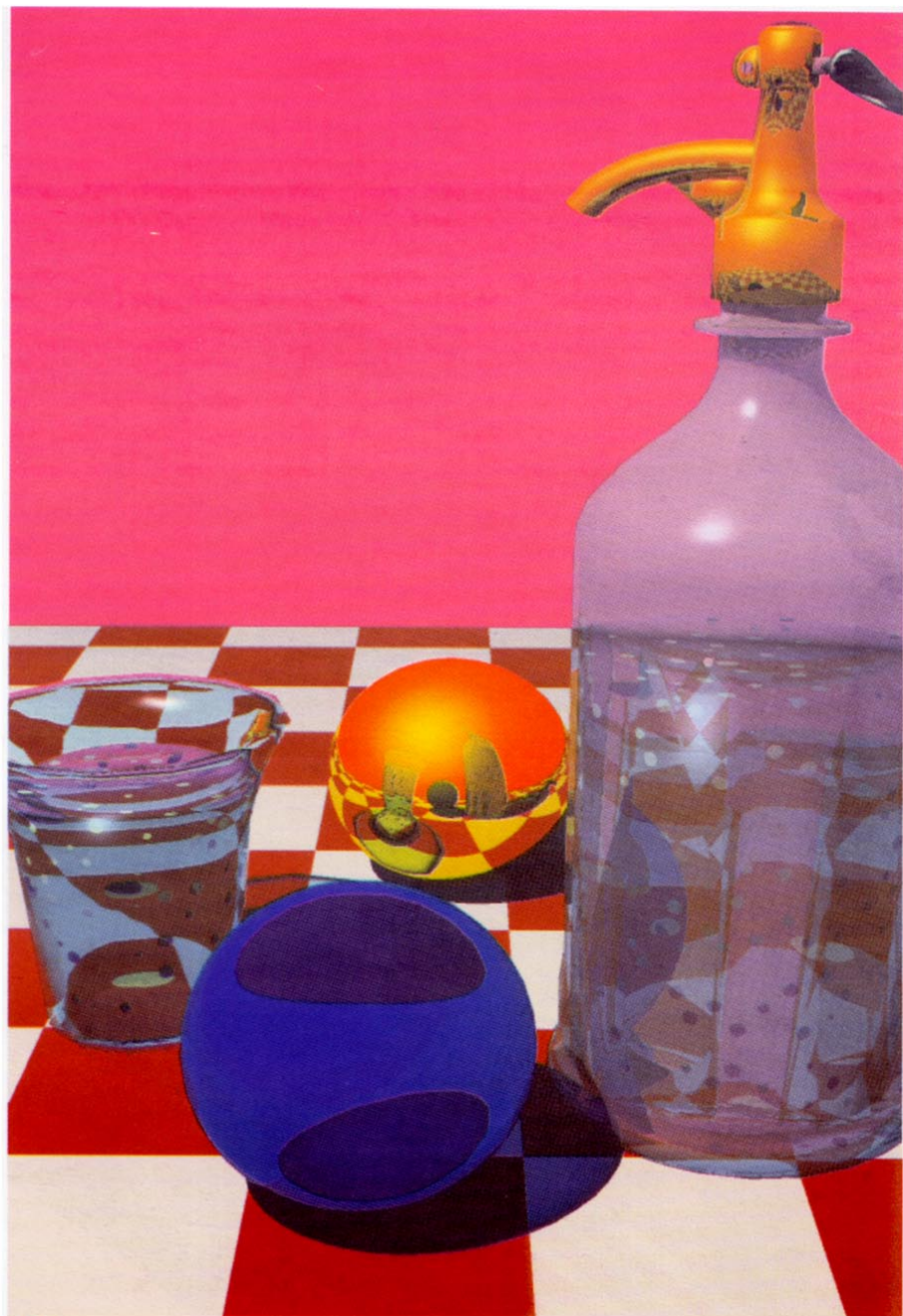
### **Boros Zoltán: Piramis**

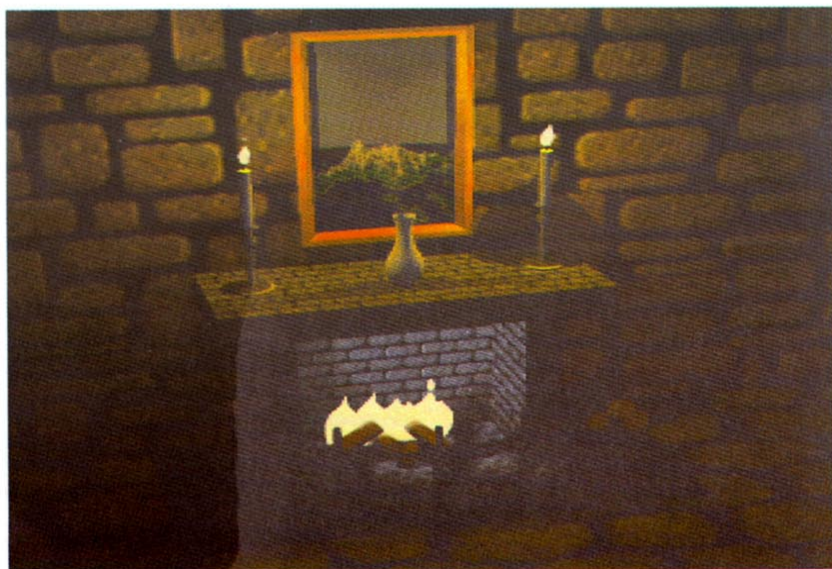
A gömböt ölelő fénykört a Lensflare effekt hozta létre. A kamera direkt a fényforrásra néz.

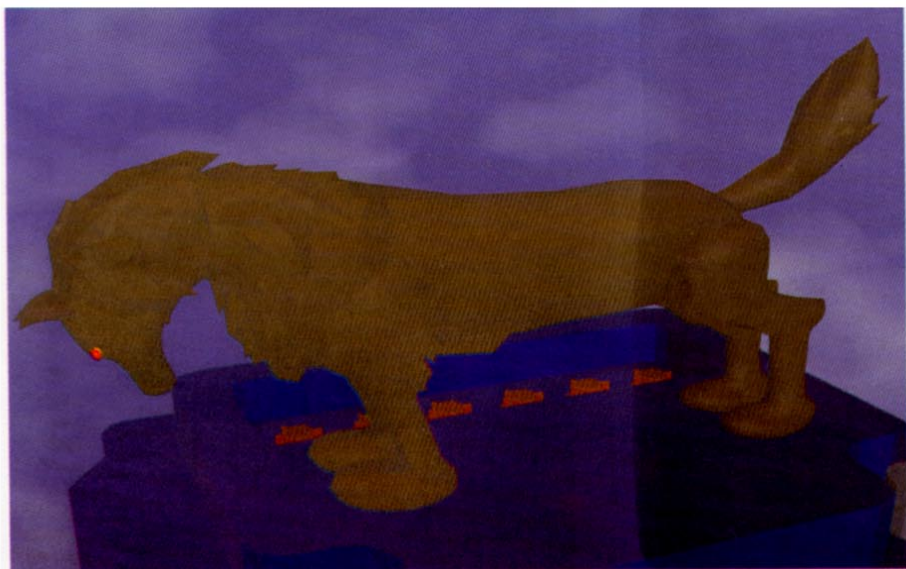


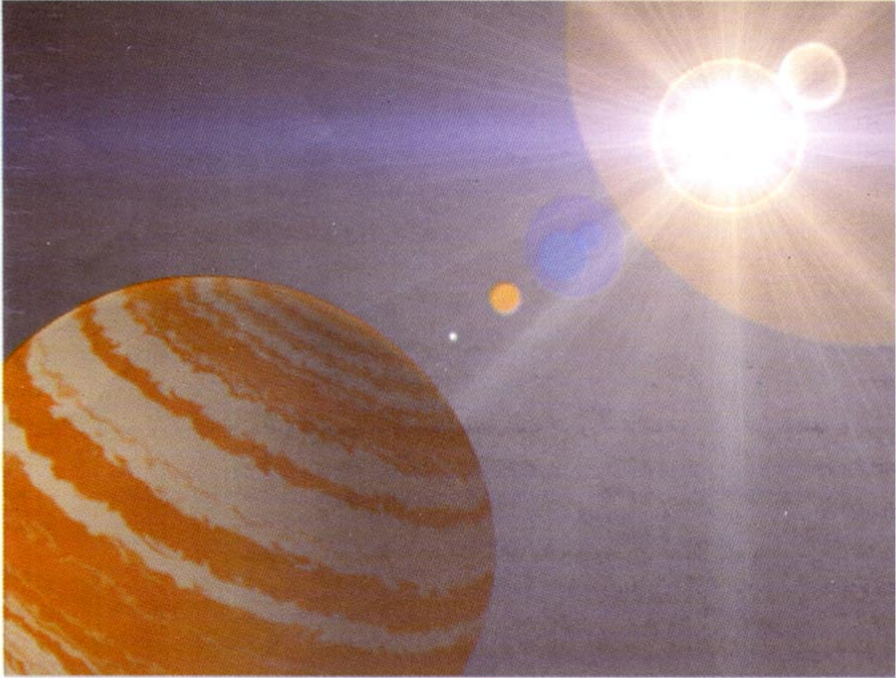
**Szívből jövő segítség**

**A véradás**

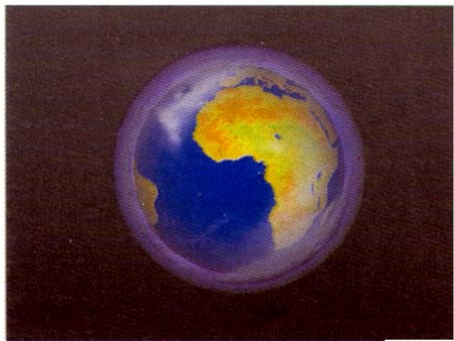








**Néhány kép az úrból**

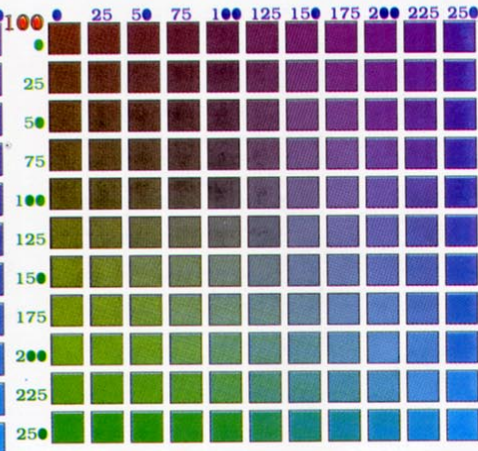
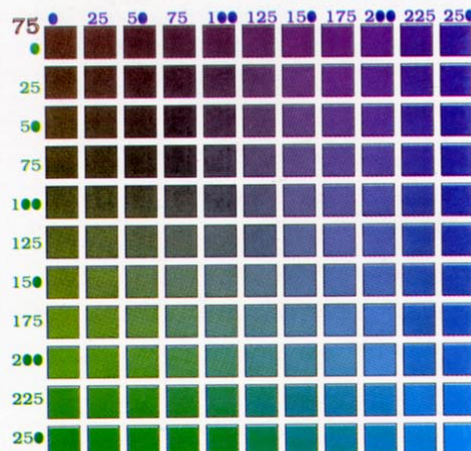
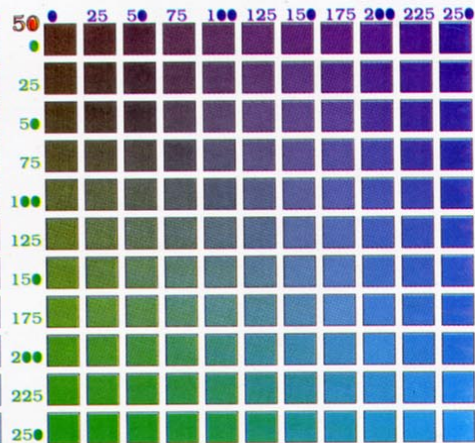
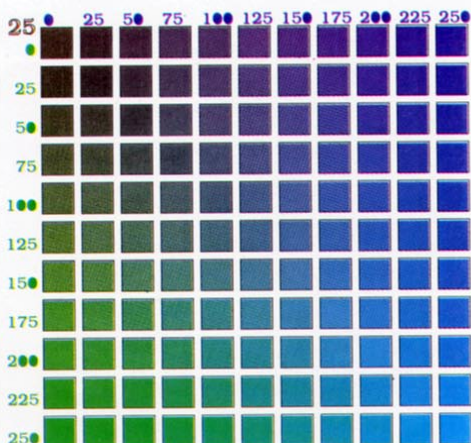
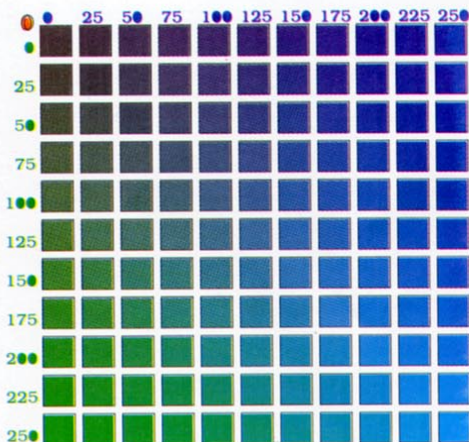


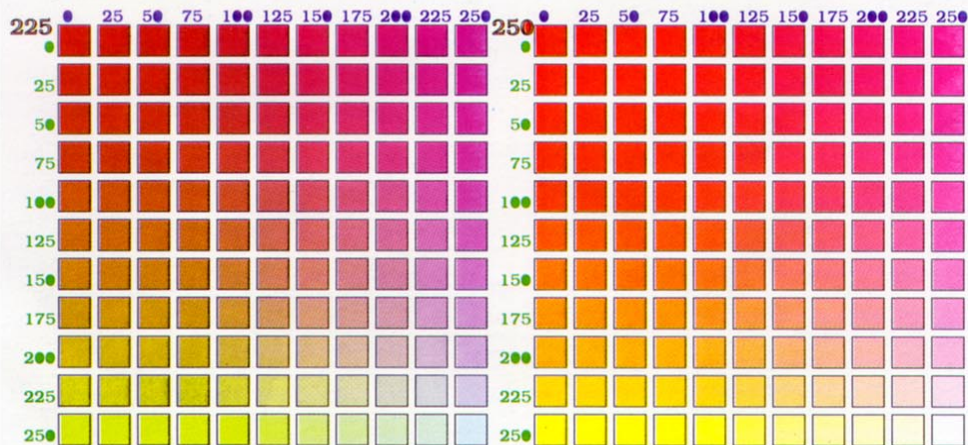
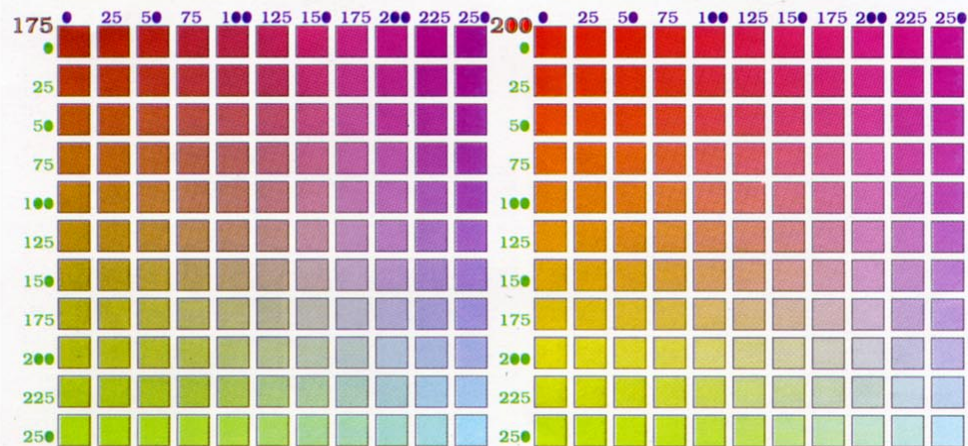
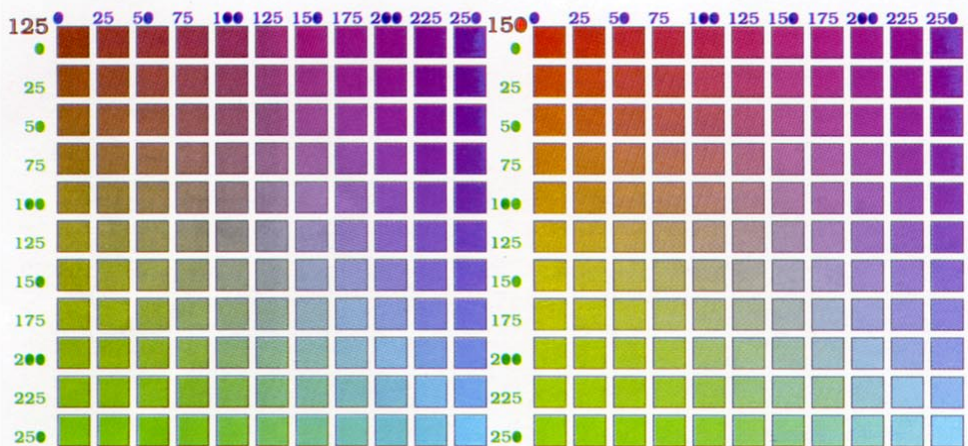


## Szintáblázat

A következő két oldalon lévő szintáblázatok az RGB összetevők 10%-os lépéseiben mutatják be a kialakuló színeket.

A kék vízszintesen a zöld függőlegesen helyezkedik el, míg a vörös táblázatonként növekszik.







Az idő lejárt

# Tárgymutató

3D hatás 75

## A;

Actor, 68  
Add Axis, 57  
Add Knots, 57, 63  
Add Layer(s), 86  
Add Points, 60  
Adobe Type 1, 55  
Align to Path, 66  
Alignment, 66  
Aperture Size, 78  
Applyque, 38  
Associate, 66  
ATPT, 49  
Attribs, 49  
Attributes, 11  
Attribútum, 46  
ATxx, 49  
AutoCAD, 6

## B;

Backdrop, 10, 85  
Bend, 35  
Big Subgroup, 29  
Bluing, 85  
Bones Subgrps, 29  
Bones Update, 31  
Break, 63  
Browse, 29, 68

## C;

Camera Lines, 83  
Cast Shadow, 52  
Check obj, 52  
Clear Backdrop, 11, 85  
Clear Zone, 10, 87  
Clipboard, 71  
Conform Group to Path, 66  
Conform Object to path, 67  
Conform to ..., 11  
Constrain, 24  
Controlled Falloff, 52  
Csontváz, 26

## D;

Deform Tool, 40  
Deformations, 32  
Delete, 63  
Diminish Intensity, 52  
Discontinuous Knot, 69  
DXF, 6

## E;

Edge Filter, 54  
Eye Separation, 76

## Tárgymutató

---

### **F;**

Field Rendering, 205  
Fill to Edge Line, 44  
FLI, 7, 11  
FLI, 80, 85  
Flip Field, 205  
FOV Angle, 9,84  
Fracture, 11, 38  
Freeze, 24

### **G;**

GIF, 7  
Global Effect, 65  
Group, 12

### **H;**

Hide, 6  
Hide Unselected, 43  
Háttérkép, 10, 85

### **I;**

IFF, 38  
IFF-Anim, 5, 7  
IFF-Animbrush, 11, 80  
IFF-ILBM, 7  
Initial Cycle Phase, 69  
Initial Direction, 57  
Ízület, 26

### **J;**

Join, 63  
Joint Lock, 24

### **K;**

Keep Y, 66  
Knot, 55, 57  
Knot Control, 58

### **L;**

Latticize, 39  
Layer, 86  
Light, 50  
Light Lines, 83  
Lightsource View, 82  
Load Backdrop, 11, 85  
Load Font, 63  
Looping, 66

### **M;**

Make Continous, 60  
Make Discontinous, 59  
Make Line, 60  
Make Quick Edges, 48  
Memory, 7  
Mold, 11  
Morfózis, 73  
Mélységélesség, 75

### **N;**

NewMode, 9, 8  
Next Brk, 81  
Next Point, 57  
No Lens Flare, 52

## O;

Object File Info, 68

## P;

Particles, 36

PCX, 7, 38

Perpective Reset, 84

Pick Knots, 63

Pick More, 53

Pinch, 35

Preferences, 49

Prev Brk, 81

## Q;

Quick Draw Toggle, 48

Quick Edges, 47

## R;

Rectangular Shape, 51

Release, 25

Release All, 25

Remove Layer(s), 87

Remove Points, 62

Reset, 9

Reset Relative, 88

Reset Views, 11, 84

Reverse Cycle Motion, 69

Round Shape, 51

Réteg, 86

## S;

Save Points, 61

Screen Distance, 76

Set Edge Line, 44

Set Zone, 10, 87

Shear, 36

Show Anim, 7

Show Pic, 7

Small Subgroup, 29

Specify Layer, 86

Spline Interpolate, 69

State, 12

Strech, 33

Strip Object, 217

## T;

Tangent Direction, 58

Taper, 34

Targa, 7, 38

TIFF, 7, 38

Twist, 34

## U;

Ungroup, 12

Unhide, 6

Unhide All, 43

Unhide Subgroup, 43

Unmake Quick Edges, 48

## V;

Vel. Scaling, 69

View from Object, 82

## Tárgymutató

---

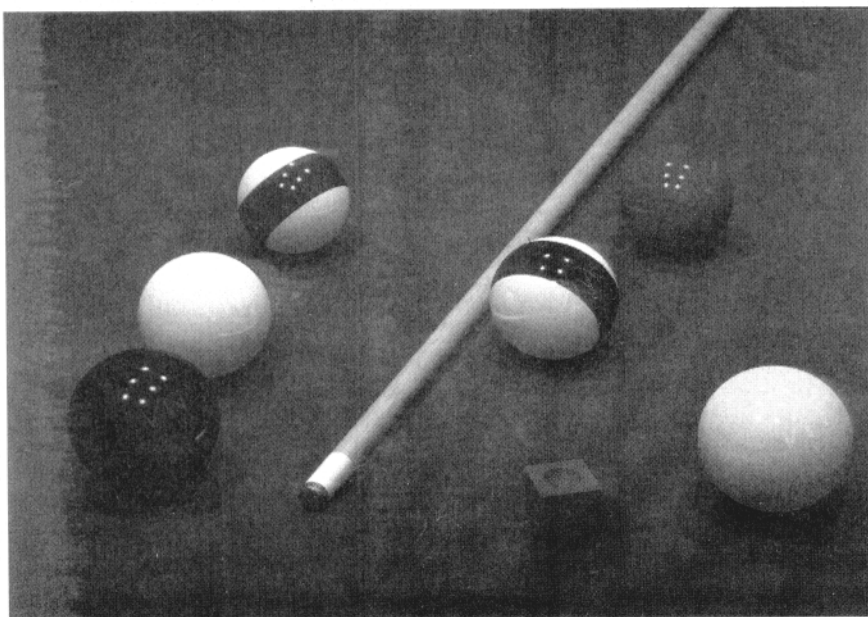
**W;**

Weave, 39

**Z;**

Zoom Ratio, 9, 84

Zóna, 87



# Tartalomjegyzék

Bevezető	3
Detail editor	5
Új megjelenítési lehetőségek	7
Régi funkciók új helyen	11
Statek	12
State trükkök	21
Inverz kinematika	22
Skeleton kontrol	26
Tárgyalakító funkciók	32
Attribútumok	46
Fényforrások	50
Egyéb újdonságok	53
Spline editor	55
Action editor	65
Tárgy betöltése	69
Beállítások másolása	71
Morfózisok készítése	73
3D hatás és mélységelesség	75
Stage editor	79
Szerkesztési háttérképek	85
Rétegek kezelése	86
Zónák	87
Végül	88
Textúrák	89
Térkitöltő textúrák	91
Agate	91
Brushed	92
Bumpnoiz	93
Colornoiz	93
Concrete	94
Confetti	95
Crumpled	96
Dinoskin	97



## Tartalomjegyzék

---

Dithered Shape család	98
Dithrect	98
Dithcirc	98
Rdditrec	98
Rdditcir	98
Dirt	99
Drtpaint	99
Easywood	100
Fakely	101
Frogskin	102
Jersey	103
Leather	103
Marble	104
Monster	104
Mosaic	105
Noiz2 család	107
Clrnoiz2	107
Refnoiz2	107
Filnoiz2	107
Oldbrick	108
Pebbled	109
Peened	110
Quilt	111
Rainbow	111
Scratch	112
Spotch	113
Staings	114
Statue	115
Terra	116
Wormvein	117
Vrinkle	118
Zoolo	119
2D textúrák	120
Bathtile	120
Blast	121
Bmpbrnch	122

Branches	123
Shingles	123
Stamped	124
Trichex	125
Tracer	126
Tritile	127
Weave	128
Alakfüggő textúrák	129
Cheks család	129
Rectchex	130
Radcheks	130
Sprlcheks	131
Deathstar család	131
Dethstar	132
Rddthstr	133
Honey Comb család	133
Honycomb	134
Radcomb	134
Hexez	134
Plaid	135
Window család	135
Radwinds	136
Rectwind	137
Tubewind	137
Speciális alkalmazású textúrák	137
Antique	138
Cndyappl	139
Coolfir	139
Craks	140
Dashline	141
Fireball	142
Gasgiant	143
Hardwood	143
Hrdstrip	144
Iris	144
Lensflar	145

## Tartalomjegyzék

---

Metals	146
Mntntop	147
Ribbed	148
Sftstrip	149
Transpar	149
Venitian	150
Zbuffer	150
Animálható textúrák	151
Beammeup	151
Cahser	153
Clouds	154
Dancspark	155
Dripdrop	156
Electric	157
Fire	158
Rain	159
Spark	160
Tiedye	161
Köd textúrák	162
Ghost	162
Fogtop	163
Fogpaint	164
Nebula	165
Light textúrák	167
Frnchwin	167
Softedge	168
Strobe	169
Venlite	170
Animációs effektusok	171
Normál effektek	171
Animbrsh	171
Baloon	172
Firewrks	174
Particle	176
Shredder	180
Spike	184

## Tartalomjegyzék

---

Sway	185
Globális effektek	186
Cepia	187
Contrast	188
Fadblack	189
Haze	189
Lensflar	190
Melt	192
Negative	193
RGBmix	195
Solarize	195
Toon	196
Tippek és trükkök	199
Falvastagság képzése	199
Kibontás a csoport mélyéről	202
Brush, textúra duplikálása	202
Mapping helyett alakzat	203
Field rendering	205
Pattogó tárgyak	207
Neonfény	209
Színes fényjáték	211
Tűzlövedék	211
Bolygók	212
A Slice hibái	215
Stripe	216
Hegyek, völgyek	217
Színes világ	219
Textúra áttekintés	220
Színtáblázat	232
Tárgymutató	235
Tartalomjegyzék	239