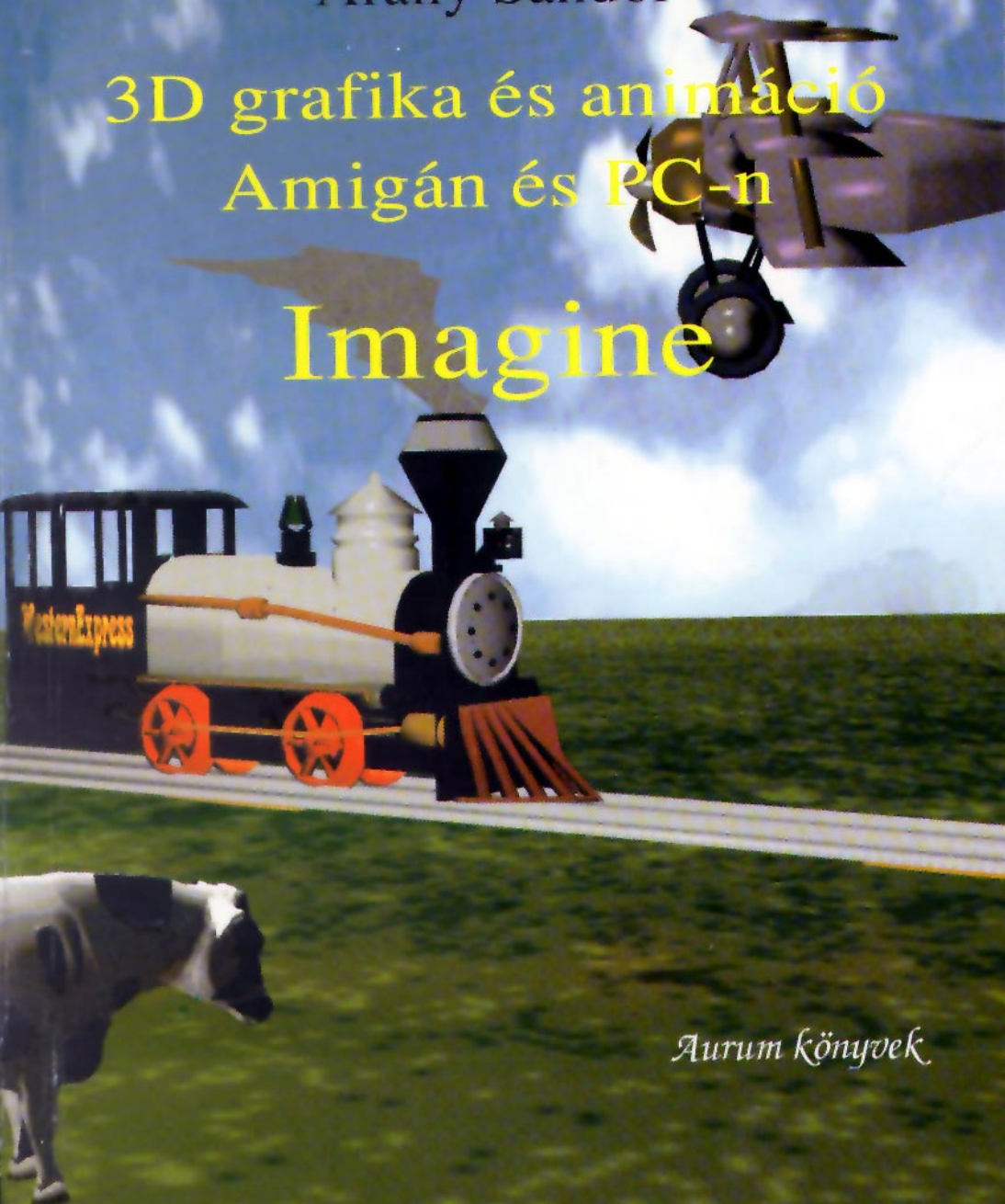


Arany Sándor

3D grafika és animáció
Amigán és PC-n

Imagine



Aurum könyvek

Arany Sándor

3D grafika és animáció
Amigán és PC-n

Imagine I. kötet

AURUM
Könyvek #4

Aurum Könyvek sorozat #4
Sorozatszerkesztő:
Arany Sándor

Lektorálta:
Marinov Gábor

3D grafika és animáció
Amigán és PC-n
Imagine I. kötet
(c) Arany Sándor 1994

ISBN 963 04 4015 6

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

Borítóterv:
Aurum

Nyomdai munkálatok:
Kollányi Nyomda Tiszaföldvár

Bevezetés

A számítógépes grafika és animáció korunkban egyre nagyobb teret követel magának. Népszerűségét elsősorban látványosságának köszönheti.

A műfaj betört a reklámokba, a film és video iparba, színes magazinokba, a hétköznapi feladatok mellett önálló művészeti ággá nőtte ki magát. Csak néhány példát sorolok. A tévéreklámok nagy része a Friderikusz Show, a Szerencsekerék, a Játék határok nélkül, de rengeteg más műsor főcíme is, a TV1 és TV2 logo, mind-mind ezzel az eljárással készültek.

A video-ipar előszeretettel használja ezt az eszközt, mivel a számítógép képe aránylag egyszerűen rögzíthető videoszalagon. Utóbbi idők video klipjei közül nekem a Pet Shop Boys "Go West"-je a kedvencem. Ez a klip végig számítógépes grafika, kitűnően ötvözve az élő képpel.

A számítógépes animáció és a mozifilm egyesítésére talán a legtöbbször által ismert példa a Terminator 2 című Schwarzenegger film.

Másik közismert példa a Disney Szépség és a szörnyeteg című rajzfilmje, amely Oscar díjat kapott a számítógépes és a rajzfilm animáció közös felhasználására. A táncos nagyjelenetben a kastély táncterme számítógéppel volt megtervezve és elkészítve, megdöbbentő élethűséggel.

A Disney stúdió annyira megkedvelte ezt az eljárást, hogy új filmükben, az Aladdinban már lépten-nyomon ezt használták.

Spielberg a Sea Quest című fantasztikus filmsozátát, amely 94 elején futot (fut) az RTL műholdas csatornán Amiga számítógépek segítségével készítették, a New Tech Light Wave 3D modellező és animációs programjával és Video Toaster nevű

Bevezetés

videografikus rendszerével. Szintén Amigán és Toasteren készült a Babilon című fantasztikus film.

Ugye kedves olvasó, nem kell tovább sorolnom a ray-tracing, a számítógépes grafika megjelenési formáit, te is tudnál példákat említeni. Nem is az a könyv célja, hogy felsorolja az összes jelentős alkotást, hanem az, hogy segítséget adjon az ilyenek elkészítéséhez.

A könyvem során nem a megszokott magázó formulát használom, hanem tegezni foglak kedves olvasó, remélem azonban ezzel nem sértelek meg. Hogy miért a pertu? Hivatkozhatnék a koromra, a többségnél idősebb vagyok vagy vele egykorú, de ezt nem teszem. Inkább arra hivatkoznék, hogy számítógépesek általában tegezni szoktuk egymást. Ez a formula közvetlenebb, barátságosabb, az ember jobban elfogadja a tanácsokat, új dolgokat a barátjától. Nem kell használni a sokszor kényelmetlen udvariassági formákat, mégsem gondolom ezt tiszteletlenségnek. Ha mégis annak érzed, úgy elnézést kérek érte Öntől.

Először talán tisztázni kellene, mi az a ray-tracing. Szó szerinti fordításban fénysugárkövetést jelent. Gyakorlatban szintén.

Egy ray-tracing kép elkészítése fázisokra bontható. Először elkészítjük a képen szereplő tárgyak három dimenziós (ez it a lényeg, a 3D!) alakját, mint egy műszaki rajzát. A tárgyakat tulajdonságokkal látjuk el, úgymint szín, fényvisszaverés, átlátszóság, felületi mintázat, stb. A kész tárgyakból a számítógépen berendezzünk egy helyszínt, ahová fényforrásokat is rakunk. Kell tenni a helyszínre (szokás színpadnak is nevezni, a rajta szerepő tárgyakat pedig színészeknek) egy képzeletbeli kamerát is.

Ha mindez kész, jöhet a fénysugár-követés. A számítógép kiszámolja, hogy ha a helyszín a

valóságban is létezne, az adott fényviszonyok mellett a kamera mit látna. Ezt úgy éri el, hogy a fény útját szimulálja, követi, egészen a kameráig, a létrejövő képet pedig rögzíti.

Ez az elméletet, gyakorlatilag máshogy történik. Mivel a fényforrásokból a tér minden irányban (kivéve az irányított fényű lámpákat) indulnak fénysugarak, amelyek nagy része soha nem kerül a kamera látószögébe, a számolás gyorsítása végett a ray-tracer programok fordítva gondolkodnak. A színpadot térrészekre osztják és először megálapítják, hogy abba a térrészbe juthatott-e fény. Ha igen, akkor a tárgy valós felületeit kell megvizsgálni, azok mely pontja kapott fényt.

Tehát a fény útját nem a lámpától vizsgálják a tárgyig, hanem a tárgytól a lámpáig, de ez csak technikai részlet.

Még így, az egyszerűsítés után is marad rengeteg számolni való. Emiatt a számításgényesség miatt a ray-tracing csak a személyi számítógépek teljesítményének rohamos fejlődésével tudott az átlag felhasználók körében elterjedni.

Összevetés képpen, a borítón lévő kép, nagy felbontásban történő kiszámolása egy Amiga 4000 típusú számítógépnek, amely másodpercenként 3 és fél millió lebegőpontos matematikai műveletet tud elvégezni, közel két órájába került. Ez durván 24 milliárd számítást jelent.

A másik gyorsítási lehetőség, hogy a program nem számolja ki a fénysugár útját, hanem különböző matematikai formulák segítségével, a felületek alakját figyelembe véve állítja össze a képet. Ez jóval gyorsabb, mint a ray-trace algoritmus, de nem annyira élethű, például igazi üveg és tükör tárgyak nem hozhatók vele létre.

Bevezetés

A fejlettebb modellező programok többféle képszámítási eljárást is felkínálnak, amiből a felhasználó kiválaszthatja az adott képnek megfelelőt.

Ha ilyen nehézkes egy kép elkészítése, mire jó ez? A számítógépen bármi megtervezhető és a képre vihető, olyan dolgok, amelyek a valóságban nem léteznek. Talán egyszerűbb lett volna visszamenni az időben és 1920 táján egy vasútvonal mellett a jó alkalomra várni egy fényképezőgéppel?

Rajzolni is elég nehéz lenne egy ilyen képet, mert a programok három dimenziós tárgyakkal számolnak, perspektivájuk olyan jó (sőt jobb), mint a fotóknak.

Ráadásul a tárgyak mozgásokat is végezhetnek. Több képet úgy elkészítve, hogy az egyes képek között a tárgyak egy kis mozgást végezzenek, animációkat hozhatunk létre. Ezeket az animációkat filmre vagy videóra véve jönnek létre a számítógépes filmek.

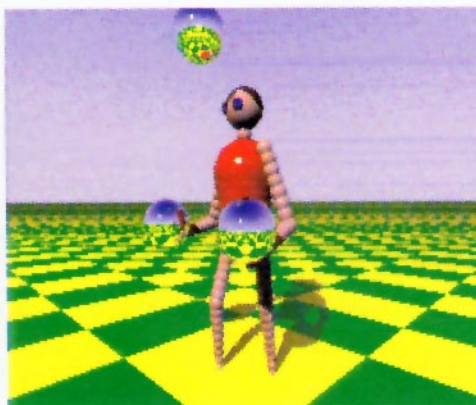
Ha minden kép elkészítése órákat vesz igénybe, mennyi idő alatt készül el egy animáció? Sok idő alatt. A sebesség érdekében engedményeket szoktunk tenni, például az animáció képkockái nem olyan nagy felbontásúak mint a nyomdai célra készültek. A borítón a kép 2400*3600 pixeles, míg video-hoz elég a 800*600 pixeles is. Ekkora méretben a kép 5-10 perc alatt elkészülhet egy gyors gépen.

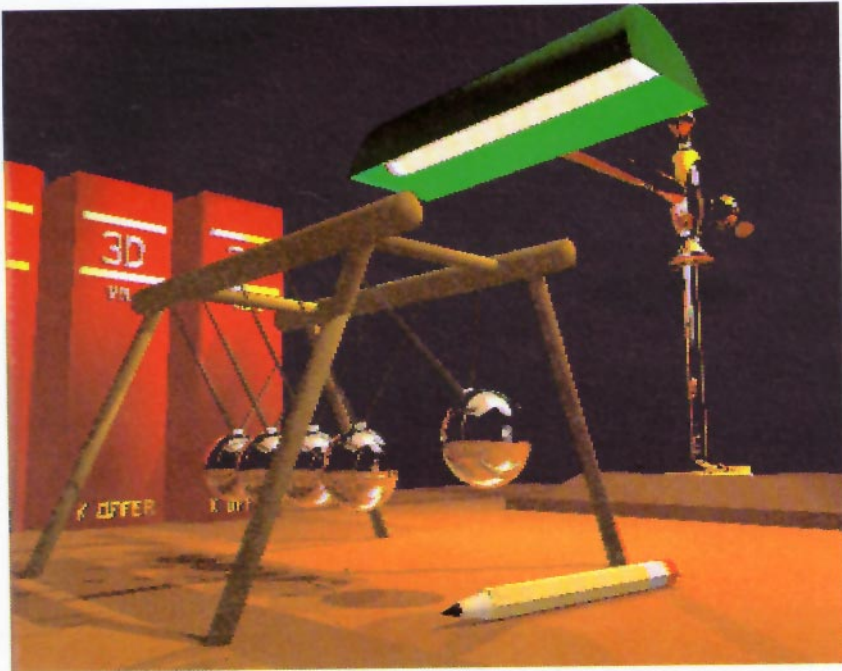
Ha az animáció film számára készül, akkor bizony nem engedhetünk a minőségből. Nem véletlen, hogy a Szépség és a szörnyetegben a ray-traced rész mindössze néhány másodperc. A Terminátor 2 számítógépes részleteit, amelyek össz időtartama 4 és fél perc, 30 darab hálózatba kötött Silicon Iris 4D/480-as gépen 30 ember készítette 10 hónap alatt. Ezekben a gépekben nyolc R3000-es processzor dolgozik, gépenként 286 MIPS és 70 MFLOPS teljesítményt nyújtva.



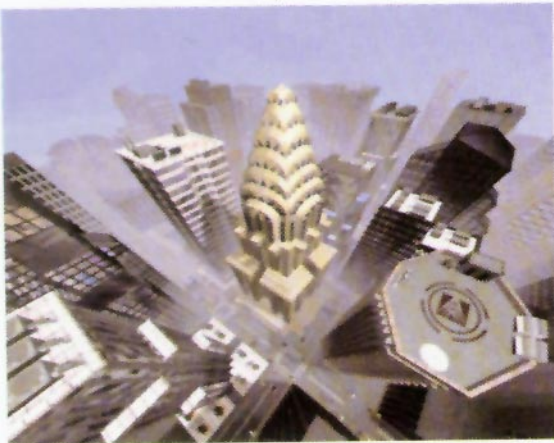
Részlet a Boingmaschine című Sculpt animációból.

Eric Graham 1986-ban készült klasszikusnak számító animációjának részlete. Az animáció a Juggler nevet viseli, Amiga 500-on készült fél mega RAM-ot tartalmazó gépen, képkockáként egy óra rendering idővel.



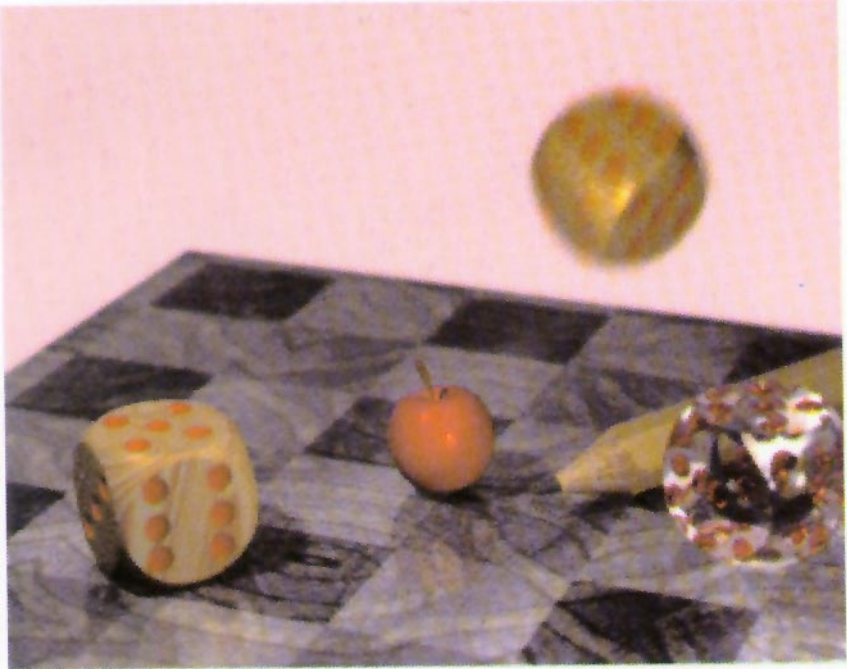


Egy másik klasszikus Sculpt animáció részlete



City

A kép PC számítógépen készült a 3D Studio programmal. A kamera fókusztávolságának csökkentésével jött létre a halszemhatás



Demókép a Real 3D V2.0-ból. A program a személyi számítógépeken futó ray-tracerek közül a legfejlettebb animációs képességekkel rendelkezik.

Vivid-del készült kép. A program szép képeket készít, de a tárgyakat és scene-t szövegfájlban kell megadni, ezért használata nehézkes, összetett tárgyak és animációk készítése szinte lehetetlen.





Részlet a Babilon című amerikai sci-fi-ből. A film számítógépes részleteit Amigával készítették a Video Toaster nevű video grafikus rendszeren a LighWave 3D modellező és animációs programmal.



A 3D Studio demóképe Jól megfigyelhető a háttérkép tükrözött csempézéses felhasználása.

Modellező és animátor programok

A ray-tracing Amigán, amely eleve egy grafikus számítógépnek lett tervezve, már nagyon sok éves múltra tekint vissza. Manapság, amikor a PC-k sebessége és grafikus képessége is jelentős fejlődésen megy keresztül, egyre több program jelenik meg erre a típusra is.

Egy röviden emlékezzünk a legjelentősebbekről. Amigán évekkkel ezelőtt, mikor még én is kezdtem, nagy népszerűségnek örvendett a Sculpt 3D, majd később a 4D. Mai szemmel nézve ezek igencsak kezdetleges programok, mégis szép képeket lehetett velük készíteni. A következő képek néhány ismertebb Sculpt animációból valók. Kritizálásukkor vedd figyelembe, hogy 85-87-ben készültek, 4096 színű, kifelbontású HAM módban. Még ma is nosztalgiával gondolok ezekre az időkre, fél megabájt RAM, egy foppy, ohne HD, 7 MHz órajel. Óh de szép volt.

89 táján megjelent egy forradalmian új szoftver, a Turbo Silver. Ez a mai Imagine elődje. 89-90 táján felváltotta az Imagine 0.99. Azóta a program sokat fejlődött, bár az általánosan elterjedt 1.1-es verzió már sok mindent tudott, abból, amit most a V2.0 tud. Időközben az Impulse bejelentette a program V3.0. változatának kiadását, egy 2.90 verziószámú előzetes változat napvilágot is látott. Ebben a könyvben csak az Imagine V2.0 programmal foglalkozom részletesen, az újabb változat taglalása a második kötet feladata lesz. Mivel a program alapvető szerkezete várhatóan nem változik meg, ezért a következő kötet csak az újdonságokkal és a megváltozott dolgokkal fog foglalkozni, a V3.0 megismeréséhez szükség lesz mindkét kötetre. Ez

Bevezetés

nem üzletpolitika, ismétlésekbe bocsájtkozni fölösleges, mikor a téma két kötetet is bőven meg tud tölteni.

Az Imagine-nel párhuzamosan kifejlődött néhány ray-tracer, ezek közül a Maxon Cinema 4D, Caligari, Aladdin 4D, Lightwave és a Real 3D V2.0 programok a legkiemelkedőbbek.

PC-n az Imagine mellett a 3D Studio és Vivid van leginkább elterjedve, de ezek egyike sem igazán ray-tracer. A 3D Studio nem igazi ray-tracing eljárást használ, de animációs képességei kiemelkedőek, sebessége is nagyon jó. A Vivid nagyon szép ray-trace képeket állít elő, de nincs tárgy és színpadszerkesztője, a képen lévő tárgyakat egy szövegfájlban kell leírni leírónyelvvvel, a sebessége pedig bőven a nagyon lassú alatt van.

94 év elején várható a Real 3D PC-s átirata is. Ez program Amigán több éves múltra tekint vissza, bár a V2.0 csak 1993 tavaszán látott napvilágot. A Real 3D V2.0 a személyi számítógépek futó programok közül jelenleg a legjobb animációs képességekkel rendelkezik, sok olyan funkciója van, amelyeket idáig csak a munkaállomások programjai tudtak.

Nem célozom ezen programok részletes ismertetése,



elég a demóképek bemutatása, azok sokat elmondanak. Elképzelhető, hogy az Aurum könyvek sorozat valamelyik későbbi kötete foglalkozni fog egymással. (Magánvéleményem szerint a Real 3D V2.0 lesz az.)



See

Demóképek a MAXON Cinema 4D-ből.



StarWars



Glasbuchstaben

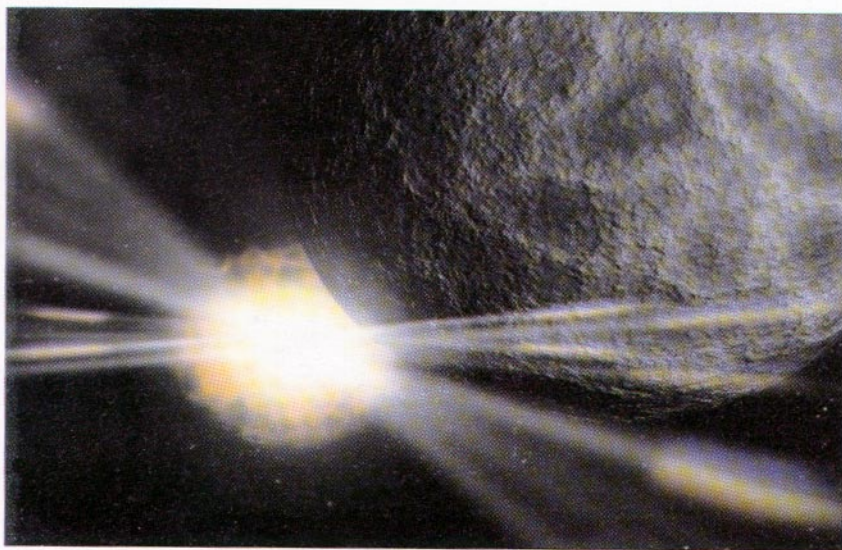
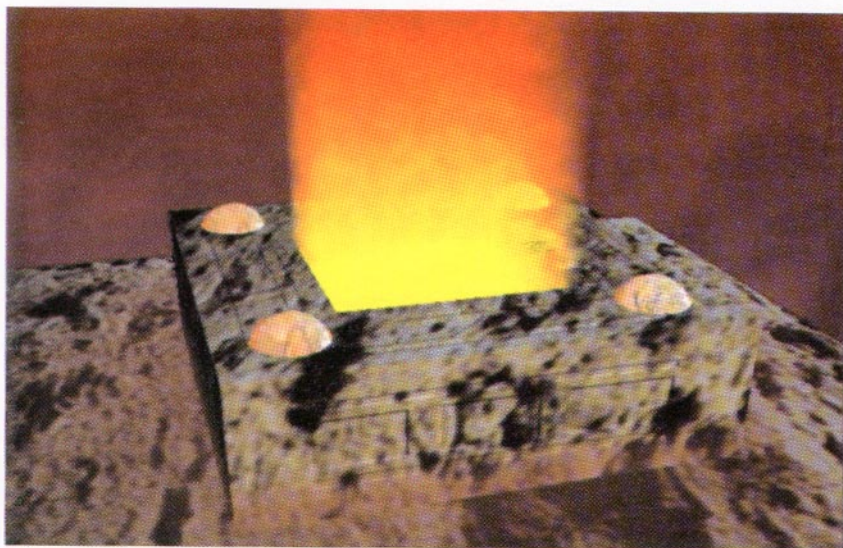
Bevezetés



Cornet

Demóképek az Aladdin 4D-ből
Discovery Galaxy







Az amerikai Newtek emblémája. Ez a cég készítette a Video Toaster-t, amely segítségével készült többek között a Babilon és a SeaQuest című film. A Toaster nem csak egyszerű 24 bites kártya, hanem komplett videofeldolgozó hardver.

Előzetes az Imagine 3.0-s változatából, amely e könyv írása-
kor még nem jelent meg. Hírek szerint a
program rengeteg
matematikai textú-
rával fog rendelke-
zni, a kép ezekből ad
ízelítőt.



A kép a már
klasszikusnak szá-
mító Caligari-
val készült. A program
erőssége, hogy nem
csak síknézetekben
szerkeszthetjük a
tárgyakat, hanem
perspektivikus drót-
vázás modelleken
is.

Általános leírás



A könyv az amerikai Impulse cég **Imagine V2.0** programját ismerteti, ezen keresztül próbál bevezetni a raytracing világába. A program először Amiga számítógépekre jelent meg, majd miután nagy sikert ért el, átírták PC-re is. Az átírat néhány

apró eltérést leszámítva teljesen azonos az eredeti verzióval, ezért együtt tárgyaljuk azokat. Mivel az AURUM DTP Stúdió csak Amigákkal dolgozik, természetesen az Amigás változat alapján készült könyv, ahol a PC verzióban eltérés mutatkozik, ott arra hivatkozni fogok.

A könyv készítéséhez az Imagine-n kívül felhasználtuk a Soft-Logic **PageStream** nevű DTP tördelőprogramját, a Great Valley Products Inc. **ImageFX 24** bites képfeldolgozó szoftverét, és a Digital Creations **Brilliance** nevű 24 bites festőprogramját. Ezen programok mindegyikét a legnagyobb magyarországi Amiga szoftverkereskedés, a **THALYMPEX Co. AMIGA-Software Distributor, 1077 Budapest, Baross tér 16** bocsájtotta rendelkezésünkre. A cég a teljes Amiga programválasztékon kívül hardver eszközöket is forgalmaz. Köszönjük.

Általános leírás

Az Amiga operációs rendszere híres a felhasználóbarátságáról és kényelmességéről. Olyan beépített kényelmi funkciókkal rendelkezik, amelyeket a futó programok kihasználhatnak, egységessé téve az alkalmazást. Hasonló elvre épül a Microsoft Windows is.

A program átiratának készítői megpróbálták ezt a kényelmi rendszert is átvenni, ezért készítettek egy **Intuition emulátor**-t, amelyen keresztül a program ugyan azokat a billentyűzetfunkciókat, kérdezőket, ablakokat és menürendszert használja mint az Amigás párja. Akik nem ismernék ezeket, azok figyelmesen olvassák át a következő részt, mivel az Amiga ablak, egér és menükezelése eltér a PC-étől.

A szerkesztő és kiválasztó műveletekre az egér bal gombja használható, míg a jobb gomb a menükezelésre van fenntartva.

Az Amiga bizonyos funkciók billentyűzet rövidítésére két vezérlőbillentyűt használ, ezekkel együtt kell használni valamely normál billentyűt. Ez a két gomb a jobb és a bal Amiga billentyű. A két billentyűt megkülönbözteti az operációs rendszer, a rendszerfunkciókra a balt használja, míg a jobbot a programok saját céljaikra vehetik igénybe. PC-n ezt a két billentyűt a két alt gombbal helyettesítették, amelyek szintén meg vannak különböztetve, a program vezérlésére a jobb oldalt használhatjuk, valamely más billentyűvel együtt.

A menüket az egér jobb gobjának lenyomásával érhetjük el. A gomb folyamatos nyomva tartása mellett fel kell vinni a pointert a képernyő tetjére, minek hatására legördül a menü. Amíg a jobb gomb nyomva van, a képernyő felső sorában a menük nevei olvashatók. A pointert a kiválasztandó menü fölé kell vinni, majd elengedni a jobb gombot, ezzel aktiváljuk a menüt. Némelyik menüpont mellett

találunk egy nagy A betűt és még egy valamilyen karaktert. Az a menüpont a jobb Amiga és a jelzett billentyű együttes használatával rövidíthető.

A program fájl kérdezői is egységesek, érdemes velük megismerkedni. A kérdező felső részén egy üzenet olvasható, általában az aktuális művelet megnevezése. Alatta találunk egy input mezőt, Drawer néven, ide kerül a kiválasztandó fájl elérési útvonala. Ezt beírhatjuk a billentyűzetről is, de a kérdező közepén lévő listából is választhatunk. A listában fájlok és direktorik is fel vannak sorolva. A tartalomjegyzékeket egy (dir) felirat jelzi. Ha direktori névre klikkelünk a bal gombbal, abba belépünk, a kérdezőben megjelenik annak tartalma, míg neve bekerül az elérési út input mezőjébe. Ha a sok fájl és tartalomjegyzék neve nem fér el a listában, a jobb oldali gördítőléccel lehet a közöttük mozogni.

A kiválasztott fájl neve kerül a File input mezőbe, billentyűzetről, vagy a listából. Az OK-ra klikkelve, vagy az enter billentyű lenyomásával fogadjuk el a fájlt. Ezzel egyenértékű, ha a listában gyors egymás utánban kétszer a nevére klikkelünk.

Ha a program valamelyik kérdezőjének input mezőibe írunk, utána mindig meg kell nyomni az enter billentyűt, a kérdező csak ekkor vesz tudomást a változtatásról. Ha ez elmarad, az input mező megtartja az eredeti értéket.

A program installálása

A program installálása igen egyszerű művelet, el lehet végezni DOS-ból, Workbenchből, vagy valamilyen fájl másoló programból. Készíts a kívánt elérési útvonalra egy könyvtárat bármilyen néven, legyen ez mondjuk Imagine, majd a program minden

Általános leírás

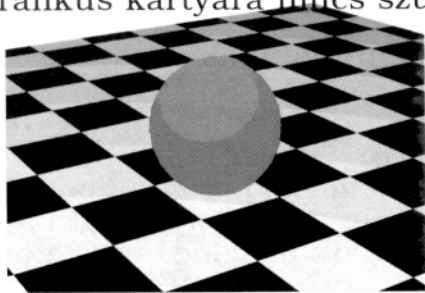
lemezének tartalmát másold fel ide. A program azonnal használható, semmilyen setup műveletre nincs szükség.

A program PC verziója két 3.5" HD-s lemezen található, egyik a programlemez, másik a demója. Az Amiga változat négy 3.5" DD-s lemezt foglal el, egyik a demo, másik az object, míg kettő a programlemez. Mindkét lemezen található egy Imagine program, ezekből csak az egyikre van szükség, attól függően, hogy milyen Amigát használunk. A kooprocesszoros gépeken az FPU nevű lemez tartalmát, míg az azt nem tartalmazó gépeken az INT lemezen lévő programot kell használni. Kooprocesszoros gépeken is használható az INT verzió, de az FPU kihasználja annak előnyeit.

A programot indíthatjuk ikonján keresztül, vagy DOS-ból nevének begépelésével.

A program hardver igénye nem túl magas, Amiga-n egy megabájt RAM és egyetlen floppy drive elég a működéséhez, kooprocesszor nem szükséges. Gyakorlatilag négy megabájt RAM és HD ajánlott. PC-n mindenképpen szükség van legalább 3 MB HD-re, 4 MB RAM-ra, 386-os, vagy nagyobb processzorra kooprocesszorra, egérre és VGA monitorvezérlőre. Az ajánlott konfiguráció 8 MB RAM, 20 MB szabad HD kapacitás, 1 MB RAM-os VGA kártya

True Color grafikus kártyára nincs szükség, de jó ha van.



Project editor

Az Imagine-t elindítva, néhány másodperc töltés után először az Imagine főcímlógóját láthatjuk. Bármelyik menüpont aktiválása után ez a kép eltűnik és megjelenik a Project editor. Ahhoz, hogy a program működését megértsük, tisztában kell lennünk az Imagine terminológiájával. Az első szakkifejezés, amivel találkozhatunk, a **"Projects"**. Ez magyarul tervet jelent. A project egy-egy munkának az elkészítéséhez szükséges teendők összessége. Minden amit a programban végzünk, általában valamilyen terv érdekében történik.

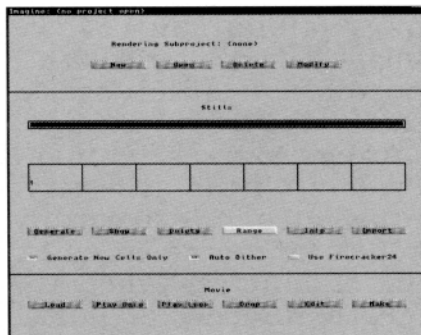
Például, ha készíteni akarunk egy képet, amelyben egy üveggolyó van egy kockás talajon, az már egy terv.

A tervekészítést felbonthatjuk három fő lépésre, a tárgyak elkészítésre, a kép vagy az animáció beállítására, majd a képek kiszámoltatására, a renderingre.

A tárgyak neve az Imagine-ben "object". Ez nem csak az üveggolyót és a kockás talajt jelentheti, hanem a fényforrást, az esetleges animációs mozgáspályát, stb is.

A kép vagy animáció beállítást **"scene"**-ek, színpadnak nevezzük. Ez tartalmazza a tárgyak helyzetét, mozgásdefinícióit, a világítást, a kameraállást és nézőpontot, stb.

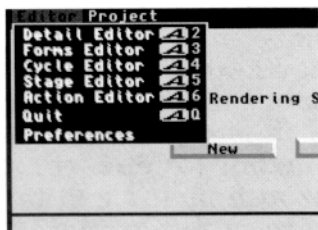
Egy terven belül nyithatunk alterveket. Ezekben közös, hogy azonos objecteket és scenét használnak



Project editor

azonban a kép mérete és leképezési eljárása különböző lehet. Ez jelentősen lecsökkenti a kép elkészítésének az idejét, mert amíg nincs kész a végső beállítás, kisebb méretű és felbontású képeket készíthetünk preview-ként, egy gyorsabb, de kevésbé részletes leképezési eljárással. Miután mindent megfelelőnek találtunk csak akkor kezdünk neki a nagy felbontású végleges kép elkészítésének. Az Imagine-ben a tervekészítés munkafázisait feladatcsoportonként külön szerkesztőkben végezhetjük. A szerkesztőket **editoroknak** is szokás nevezni.

Az első szerkesztőt már láthatjuk a képernyőn, ez a **Project editor**. Itt a tervet kezelhetjük, újat hozhatunk létre, alterveket nyithatunk, elindíthatjuk a képek kiszámítását és az animációt is itt készíthetjük el. Mindezek mellett a kész képek és animációk megtekintésére is van lehetőség. Mielőtt részlete-



sebben megismerkednénk a Project editorral, röviden tekintsük át a többi editor feladatát. Az egyes szerkesztőkbe átlépni a Project editorból az Editors menü pontjainak kiválasztásával lehet. A többi szerkesztőben ezek a menüpontok a Project menüben található.

Detail editor - A tárgykészítés szempontjából a legfontosabb szerkesztő. Ebben tárgyakat hozhatunk létre, beállíthatjuk azok attribútumait, úgymint szín, csillogás, fényvisszaverés, felületi mintázat, stb.

Forms editor - Ez is egy tárgyszerkesztő, de itt nem a részletes kidolgozás a feladat, hanem összetett tárgyak durva, előkészítő jellegű kimunkálása. Az itt készített objekteteket a Detail editorban még tovább szoktuk finomítani.

A Project editor

Cycle editor - Ebben a szerkesztőben a már elkészített tárgyakból állíthatunk össze ciklikusan mozgó egységeket, mint például egy sétáló alak, vagy egy gőzmozdony tolattyúí. A létrehozott ciklikus object az animáció folyamán automatikusan fogja végezni az itt beállított mozgásokat.

Stage editor - A kép vagy animáció színterét szerkeszthetjük ebben az editorban. Itt helyezhetjük el a tárgyakat, a kamerát itt állítjuk be, stb.

Action editor - Ez is a stage azaz a színpad beállításainak elvégzésére szolgál, de itt nem interaktívan, grafikusán, hanem számszerű értékek megadásával, jóval pontosabban végezhetjük ezt el. Főleg az animációk beállítására használjuk.

Preferences - Ez a szerkesztő nem tartozik szorosan a project-hez. Ebben magának az Imagine-nek a beállítására van lehetőség. Itt állíthatjuk be a képernyő színeket, a fő elérési utakat és sok egyéb dolgot.

Miután nagy vonalakban megismerkedtünk a program editoraival, kezdjük neki a Project editor részletes megismerésének. Ehhez először hozzunk létre egy új tervet.

A **Project menü New** pontjának kiválasztása után egy fájl szelektort kapunk, ahol a terv nevét és elérési útvonalát kell megadnunk. Legyen ez például Terv. Az elérési utat természetesen válassza ki magának mindenki. Már van egy megnyitott tervünk, nézzük meg, hogyan néz ez ki a lemezen.

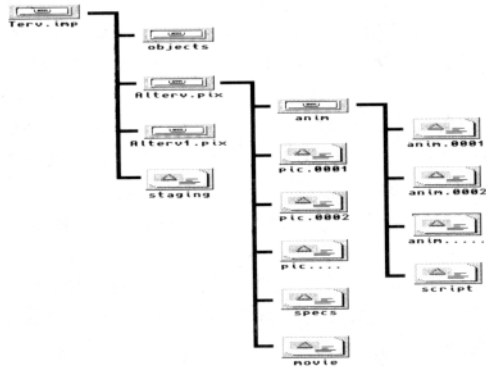
Az előző műveletünk létrehozta a megadott útvonalon a **Terv-imp** tartalomjegyzéket, abban az **objects** altartalomjegyzéket és a **staging** fájlt.

Az objects-be tehetjük a későbbiekben az elkészített tárgyakat. Nem föltétlen szükséges a tervhez tartozó tárgyaknak ebben a tartalom-

Project editor

jegyzékben lenni, de célszerű, mert így együtt kezelhetők, például másolásakor vagy archiválásakor.

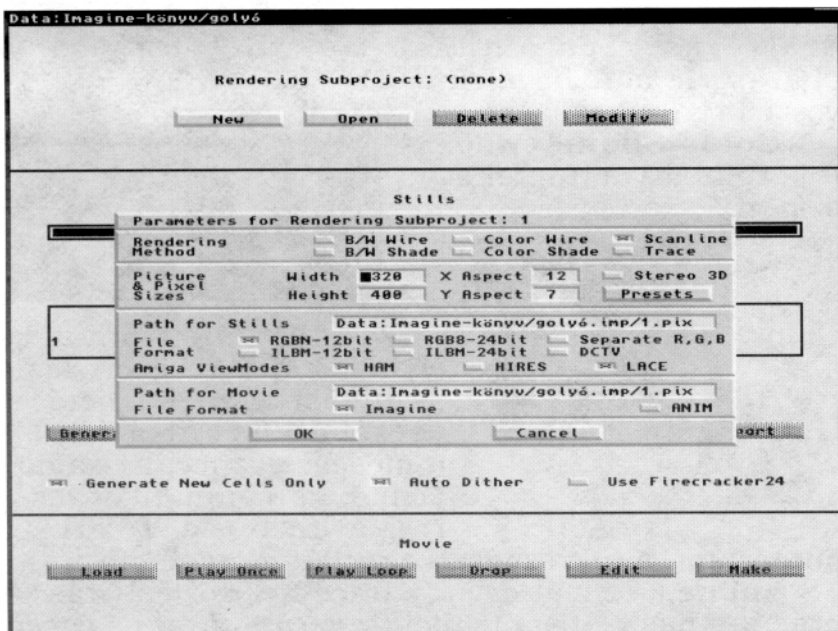
A képen a terv tartalalmjegyzék struktúrája látható. A feltüntetett fájlok és tartalomjegyzékek némelyike csak bizonyos esetekben jön létre (például az **anim** csak animáció készítésekor).



A staging a színpadi beállításokat tartalmazza. Alap helyzetben csak egy kamera van a színpadon. Miután létrehoztunk egy új tervet, vagy az **Open** menü ponttal megnyitottunk egy már létezőt, a terv elérési útvonala és neve megjelenik a képernyő felső sorában, az állapot sorban. A későbbiek során is, ebben a sorban fontos információkat fog velünk közölni a program.

Ahhoz, hogy a terven munkát végezzünk, altervet is kell létrehozni. Ezt a **Rendering Subproject:(none)** alatti négy gadget közül a **New**-val tehetjük meg, ahol az alterv nevét és útvonalát adhatjuk meg. Bár a program megengedi, hogy az altervet tetszőleges útvonalon hozzuk létre, célszerű azonban elfogadni az alap beállítást, így az alterv a terv által létrehozott tartalomjegyzékben jön léte.

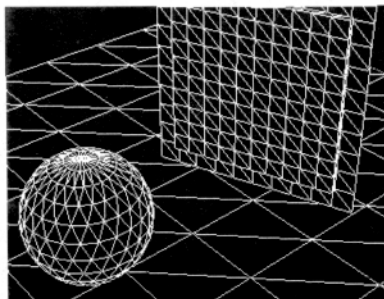
Legyen az alterv neve például preview. Miután beírtuk a nevet és elfogadtuk azt az **OK**-val, vagy az enter leütésével, egy sok beállítási lehetőséget tartalmazó kérdezőt kapunk, ahol a subprojectben előállítandó képek és animáció paramétereit adhatjuk meg. A kérdező felső sorában az alterv neve



olvasható. Alatta **Rendering Method** néven hat kapcsolót találunk. Ezek közül az egyik be van kapcsolva. Ezt a négyzetben megjelenő **X** mutatja. Ennek a hat kapcsolónak a funkciója a leképezési módszer kiválasztása. Tekintsük át az egyes módszereket részletesen.

A legelső, a **B/W Wire** a legegyszerűbb és leggyorsabb képszámítási módszer. Ez mindössze egy fekete-fehér kitakart drótvázis képet számoltat ki.

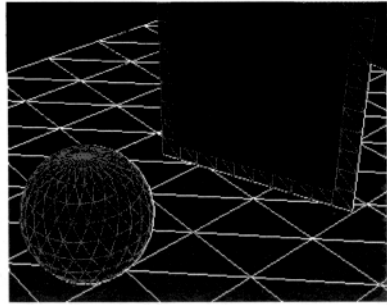
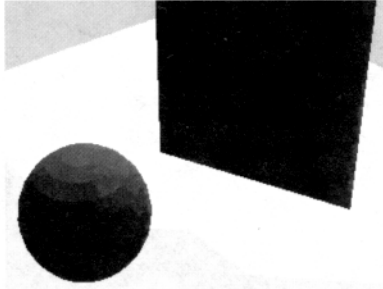
A **B/W Shade** szintén fekete-fehér képet eredményez, de itt már a felületelemeket is árnyalja a program.



Project editor

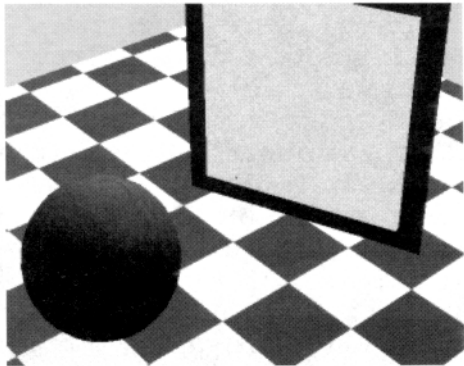
A **Color Wire** ismét egy drótvázás képszámítási eljárás, de már figyelembe veszi a tárgyak színét is.

A **Color Shade** ennél is fejlettebb, a felületeket is színezi.

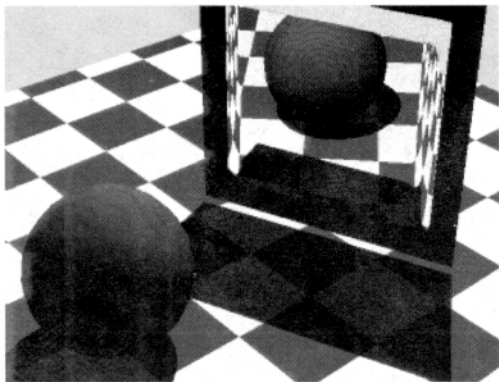


mintázatot, a tükröződést, az átlátszóságot, stb.

Scanline - Ez már egy ray-trace-hez közeli eljárás. A program használja a felületi szín, mintázat és egyéb tulajdonságokat, de a tükröződést nem, helyette csak fényfoltokat helyez el a tükröződő felületeken. Kivételt képez ez alól a háttér brush, amely már ebben az esetben is megjelenik a felületeken. A program már Scanline módban is számol az átlátszósággal, fénytöréssel azonban nem, valamint a tárgyak által vetett árnyékokat sem készíti el. Minezek ellenére sok esetben elegendő képszámítási mód, amely jóval gyorsabb, mint egy valódi ray-trace eljárás.



Trace - Az utolsó kapcsoló a valódi ray-tace képszámítási metódust kapcsolja be. Ez a leglassúbb, de legszebb képet eredményező eljárás. Itt a program kiszámolja a tükröződő felületekben létrejövő képet, az átlátszó tárgyak fénytörését, a vetett árnyékokat,



stb. Ezért a részletességért azonban nagy árat kell fizetni, lassabb gépen akár több órát is igénybe vehet egy-egy kép elkészítése, szemben a Scanline néhány percével. Nincs is mindig szükség erre a részletességre, sokszor elegendő az előző

módszer. A trace algoritmusra jellemző például, hogy két szembenéző tükröződő felületben többes tükröződést is számol.

A Rendering Parameters kérdezőben a következő egység a képméret beállítására szolgál. Itt megadhatjuk az input mezőkben a kép szélességét, magasságát pixelben, a pixelek oldalarányát, ami a megjelenítő eszköztől függ, valamint, hogy 3D Stereo szemüveggel szemlélhető képet akarunk-e. Ez utóbbi hatására olyan képet készít a program, amelyet egy speciális szemüveg segítségével valóban három dimenziósnak látunk.

A kép méretét és a pixelarányt nem szükséges manuálisan beírni ugyanis a Presets kapcsolóra klikkelve egy listát kapunk, ahol előre definiált beállítások közül választhatunk. Ezt a listát egy későbbiek során ismertető módon meg is változtathatjuk, beírva oda a kedvenc képfelbontásainkat.

Project editor

A következő egység a kérdezőben a képfórmátum beállítása. Itt egy input mezőben megadhatunk az alap elérési útvonal helyett más utat is. A kapcsolók jelentésben van némi eltérés a program Amiga és PC változatai között. Mindkét program a neki megfelelő géptípuson elterjedt képek közül enged választani. Amigán ez lehet RGBN-12bit, vagy RGB8-24 bit, amely az Impulse saját 12 és 24 bites képfórmátuma, vagy ILBM-12, 24 bit, amelyek a szabványos IFF-ILBM 12 és 24 bites képeket jelentik. Ezen kívül van még két speciális fórmátum, a Separate R, G, B, amely egy 24 bites fórmátum, de a képeket három színösszetevőre bontva, külön-külön 8 bites fájlokban tárolja el. A másik a DCTV, amely az Amiga egy pszeudo 24 bites megjelenítőjének saját fórmátuma.

A program PC-s átíratában TIF, GIF és PCX képtípusok között választhatunk.

Ennek a résznek az alján található még három kapcsoló, a HAM, a HAM fórmátumot kapcsolja be, a HIREs a nagy felbontást, a LACE pedig az interlace üzemmódot. Ezeknek csak Amiga-n van jelentősége.

A kérdező utolsó egységében az animációs fájl elérési útvonalát és fórmátumát lehet beállítani. Az útvonalat érdemes az alapon hagyni, hisz ez is a létrehozott project tartalomjegyékén belülre mutat.

Az animációs fájl fórmátuma Imagine vagy Anim lehet. Előbbi egy saját fórmátum, lejátszani a **PlayIANM** programmal lehet, utóbbi az elterjedt **Anim-5-öt** takarja. PC-n az Anim fórmátum helyett **Flick**-et választhatunk, amely a **Flick anim** animációs fórmátumot jelenti.

Válasszunk most egy nekünk megfelelő beállítást, majd klikkeljünk az OK-ra. Ezzel létrehoztunk egy altervet, aminek a neve megjelenik a "Rendering Subprojec:" után.

Ha a későbbiek során egy már létező altervet akarunk megnyitni, azt az **Open** kapcsolóval tehetjük meg, egy szokásos fájlkérdezőn keresztül.

A főlegessé vált alterveket megnyitásuk után a **Delete** kapcsolóval törölhetjük.

Ha az előzőleg tett beállításokat később módosítani szeretnénk, azt a **Modify**-ra klikkelve tehetjük meg. Ez visszahozza az imént tárgyalt paraméterkérdőzt. Arra azonban figyelemmel kell lenni, ha megváltoztatjuk a rendering paramétereket, akkor az adott subproject-hez tartozó, már elkészült képek törlődnek.

Miután kialakítottuk a nekünk megfelelő subproject paramétereket, továbbléphetünk a program megismerésében. Az újonnan megnyitott altervünk tartalmaz egy alap színpadi beállítást, amely egyetlen képkockányi. Azt, hogy a tervünk hány képkockából áll, a Project editorban is megtudhatjuk. A képernyő közepén lévő sávban láthatjuk az egyes képkockákat jelképező számokat. Most itt csak az 1-es áll. Egyszerre 70 kocka fér el ebben az ablakban. Ha az animáció ennél hosszabb, a Stills alatti tolókával lehet lapozni közöttük. Ha egy kép már ki van számolva, a neki megfelelő szám alatt egy csillag áll.

Az egyes képkockákkal való műveletek elvégzésére szolgál az ablak alatti hat nyomógomb. Hatásukat a kijelölt képkockákon végzik el. Képkockát kijelölni kétféle képpen is lehet. A legegyszerűbb, ha az azt jelképező számra klikkelünk. Több kép kijelölésekor a Shift billentyűt nyomva kell tartani.

A másik megoldás a **Range** kapcsoló. Erre klikkelve egy kérdező nyíl, ahol a kijelölni kívánt első és utolsó képkockát, valamint a lépésközt kell megadni, majd lenyomni az entert. Ez a kijelölés oda-vissza működik, azaz már kijelölt kockák, ha belesznek a sorozatba, elvesztik kijelöltségüket.

Project editor

A kijelölt kockákkal többféle műveletet végezhethünk, vegyük sorba ezeket.

Generate - Elindítja az adott képkockák kiszámítását. A képszámítás elvégzését az alatta lévő "**Generate New Cells Only**" kapcsoló állása is befolyásolja. Ha ez ki van ikszelve, csak azon képek generálását indítja, amelyek még nincsennek kész. Egyszer már kiszámított képkockát csak ezen opció kikapcsolása után tudunk újragenerálni.

Show - Megjeleníti a kiszámolt képet. Ha a kép 24 bites, de a megjelenítő eszköz nem az, akkor az "**Auto Dither**" kapcsolótól függően alkalmazza vagy sem a ditheringet a megjelenítéshez. Ez egy olyan eljárás, ami megpróbálja összemosni az egymás melletti színeket a jobb színátmenet elérése végett.

Delete - Törli a kiszámolt képeket, ezután a Generate akkor is újraszámítja őket, ha a **Generate New Cells Only** be van kapcsolva.

Info - Rövid információt ad egy már elkészült képről. Az információ kiterjed a kép méretére, készítésének dátumára, és a kiszámításához szükséges időre (**rendering time**).

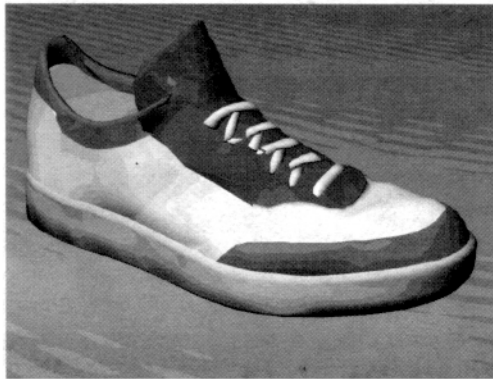
Import - Ezzel a kapcsolóval képeket importálhatunk az animációs sorozatba. Például, ha az animáció digitalizált képeket is fog tartalmazni, azokat bemásoljuk az alterm képeinek elérési útvonalára Pic.xxxx néven, ahol az xxxx a képek sorszámja, majd kijelölve azokat, az Import-tal importáljuk. Ez annyit csinál, hogy "megcsillagozza" őket, a program úgy veszi, mintha azok már ki lennének számítva. A képmanipulációs kapcsológombok alatt találunk még három kétállású kapcsolót is. Kettőről már volt szó, a harmadik, az **Use Firecracker 24** bekapcsolása után a program használni fogja az Impulse true color megjelenítőkártyáját (persze csak ha van ilyen a gépben).

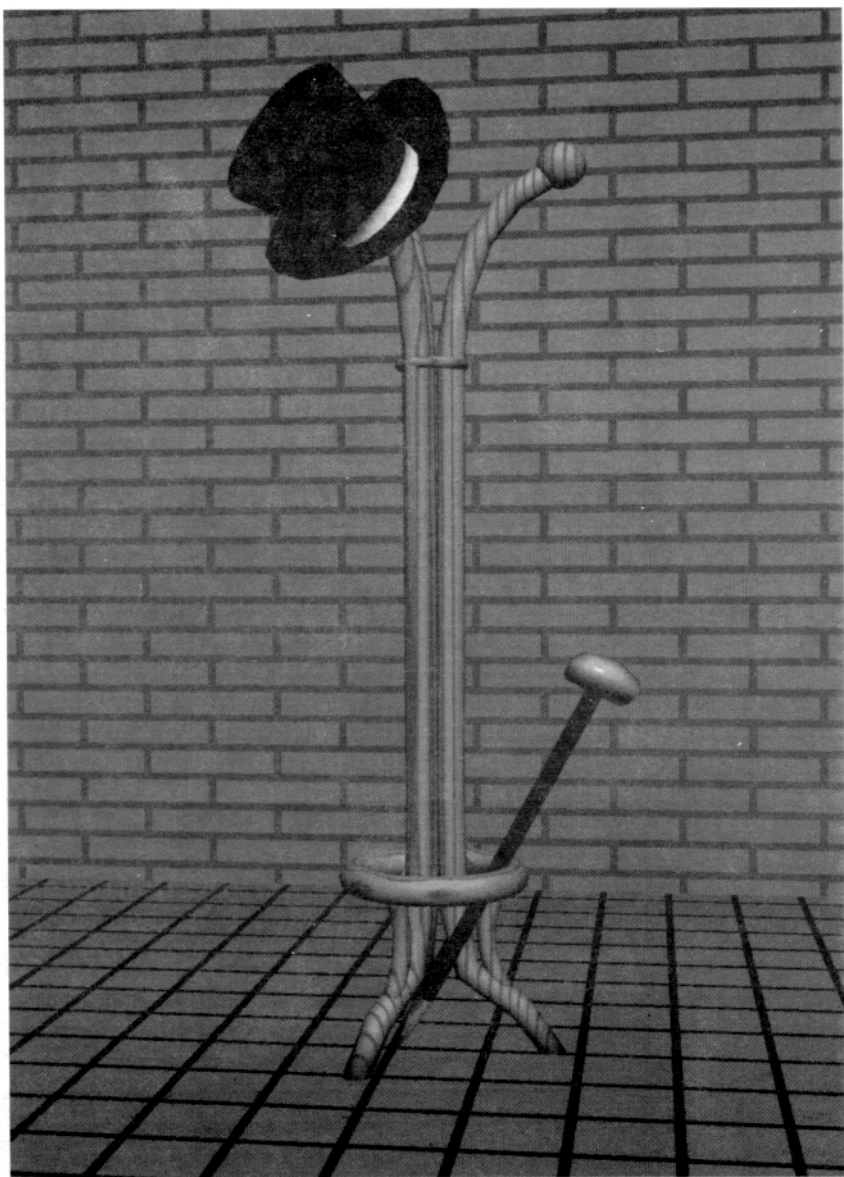
Project editor

A képernyő alján, a **Movie** fejlécű részben az animáció összefűzésével és lejátszásával kapcsolatos funkciógombok találhatók. Ezek közül a **Load** betölti a memóriába az altervnek megfelelő animációt, a **Play Once** ezt lejátsza egyszer, a **Play Loop** pedig ismétlődve. A **Drop** törli az animációt a memóriából. Utóbbi három gomb csak akkor használható, ha előzőleg már betöltöttük az animációt a Load-dal. Az Imagine csak az Imagine formátumú animációk megjelenítésére képes, az Anim-5-re és a Flick anim-ra nem.

Az **Edit** az animáció összefűzését vezérlő fájl szerkesztését indítja, a **Make** pedig ezen fájl alapján összefűzi az animációt. Részletes ismertetésükre később kerül sor.

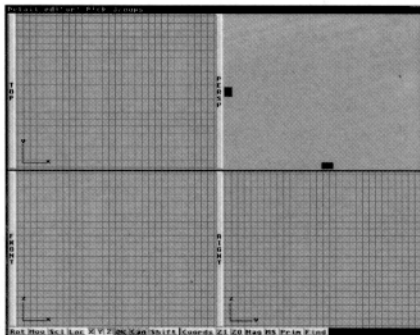
Ezzel gyakorlatilag végeztünk is a Project editor ismertetésével, bár még vissza fogunk ide térni, amikor az animációk készítéséről esik szó. A program megismerését folytassuk a legfontosabb tárgyszerkesztőben a Detail editorban. Ide, mint már említettem, az Editor menü első pontjának kiválasztásával léphetünk át. Ezzel egyenértékű a jobb oldali Amiga billentyű (PC-n ezt az Alt helyettesíti) és a 2-es egyidejű lenyomása.





Detail editor

A **Detail**-ba átlépve egy négy ablakos szerkesztő képernyő tárul elénk. Ez a négy nézetes rend jellemző az **Imagine**-re, a **Detail editor**on kívül ezt használja a **Forms**, a **Cycle** és a **Stage editor** is. Mielőtt belemélyednénk a **Detail** részleteibe, tekintsük át a program - tárgyszerkesztőinek alapkonceptióját és az **Imagine** térbeli világát. Az itt leírtak tehát azonosak (kisebb, nem lényegi eltéréseket leszámítva) a **Detail**, **Cycle**, **Forms** és **Stage editor**okban.



Az **Imagine** általános felépítése

Az editorok közötti átlépést a **Project menü** megfelelő pontjaival lehet kezdeményezni. Most tekintsünk el ezek kipróbálásától, rövidesen mindegyik sorra fog kerülni.

Az editorok nevén kívül, más menüpontokat is találunk a **Project** menüben. A **Quit** gondolom nem szorul részletes magyarázatra a programból való kilépésre szolgál. Ha az aktuális változtatások nincsenek kimentve a program mind az editorváltás, mind a kilépés előtt biztonsági kérdést tesz fel, csak azután hajtja végre a műveletet.

A következő menüpont az **Undo**, a legutolsó szerkesztő műveletet vonja vissza. Ha ez a művelet egy másik szerkesztőbe átlépés volt, nem fog működni!

A **Preferences** egy újabb szerkesztőt nyit meg, de a

Detail editor

jelenlegiből sem lép ki. A preferencest elhagyva ott folytathatjuk a munkánkat, ahol abba hagytuk. A preferencias a program működési paramétereinek a beállítására szolgál. Itt lehet például a képernyőszíneket, vagy az **User gadget**eket megváltoztatni. A részletes ismertetésre a könyv végén kerül sor, addig is a program gyári beállításait használjuk.

A **Quick Render** az Imagine egy nagyon hasznos szolgáltatása, a perspektíva ablakban látható képbeállítás, a preferencias-ben beállított formátumban le rendereli. Ehhez nem szükséges sem a kamerát sem a lámpákat beállítani, gyorsan és könnyedén kaphatunk a tárgyainkról előzetes képet.

A **Stage editor Project** menüje tartalmaz még egy pontot, a **Save Change**-t, amely a beállítások elmentésére szolgál. Erről részletesen a Stage editornál olvashatsz.

Megjelenítés

A program általános koncepciójához tartozik a **Display** menü is. Ebben a megjelenítéssel kapcsolatos dolgok szabályozhatók. Mivel ezek sem bonyolult, vagy nehezen érthető dolgok, elegendő röviden megemlíteni azokat.

A Display menü megértéséhez meg kell ismernünk a megjelenítés elvét, a képernyő felépítését. A képernyő tetején most is az állaposort találjuk. Ebben látható a szerkesztő neve, a szerkesztési mód, többes kijelölés esetén a kijelölt tárgyak száma, de itt jelenik meg adott esetben a pointer koordinátája is, stb.

A képernyő alján különböző kapcsolókat találunk. Ezeket **User Gadget**eknek nevezzük, ugyanis a felhasználó a preferenciasen keresztül maga határoz-

hatja meg a feliratukat és a funkciójukat. Egyenlőre fogadjuk el a program alapbeállításait.

A képernyő legnagyobb részét maga a szerkesztő terület teszi ki. Ez most négy ablakra van osztva. Bal felül a **Top view**, felülnézet, alatta a **Front view**, az előlnézet jobbra lent a **Right view**, a jobb oldalnézet található. Az ablakok neve a bal oldalukon lévő sárga oszlopba van írva. Ezekben az ablakokban szerkesztési segédrácsot találunk, ez azonban lekapcsolható a **Display menü Grid On/Off** pontjával. Ha a menü előtt egy pipa jel található, a rács be van kapcsolva. A rács sűrűségét is beállíthatjuk, ehhez a **Display menü Grid Size** pontját kell aktiválni. A rács sűrűsége vízszintes és függőleges irányban mindig azonos, értéke **Imagine egységben** van megadva.

Mi az az **Imagine egység**? Ez egy teljesen önkényesen megválasztott hossz mérték a programban. Mivel minden méretet és távolságot ebben a mértékegységben adunk meg, így a tárgyak modellezése során bármilyen egységgel helyettesíthető, például cm-rel, mm-rel, stb.

Pontosabb szerkesztési munkálatokhoz nem mindig elegendő a negyed képernyő, szükség volna a teljes képernyőre. Nos az Imagine-ben van erre mód, a kinagyítani kívánt ablak nevére kell kattintani a bal gombbal. Ennek hatására az adott ablak teljes képernyős lesz. A neve most is a bal oldalán található, a másik háromé pedig egymás alatt jobbról. Ha ismételten kattintunk az ablak saját nevére, visszatérünk a négy nézetes rendhez, ellenben ha jobbról egy másik ablak nevére kattintunk, az lesz teljes képernyő nagyságú.

Nem esett szó a negyedik ablakról, a **Perspective view-ről**. Ebben nem találunk segédrácsot, ez ugyanis a perspektíva ablak. Ebben, a másik három ablaktól függetlenül, bármelyik irányból megtekint-

Detail editor

hetjük a tárgyak perspektivikus képét. Az ablak alsó és bal oldali részén található egy-egy kis fekete négyzet. Ezekkel tudjuk vízszintes és függőleges irányban forgatni az ablakban lévő képet, így a tárgyak minden irányból megszemlélhetők.

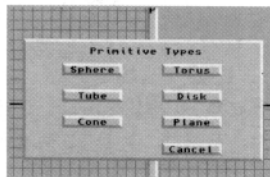
A térbeli helymeghatározáshoz szükség van egy koordináta rendszerre. Az Imagine koordináta rendszere úgy helyezkedik el, mint a geometriában használatos térbeli koordináta rendszer, vagyis az X a vízszintes tengely, melynek számozása jobbra növekszik, Z a magasság, felfelé növekvően, Y pedig a mélység, távolodva növekedve. Azt, hogy az egyes nézetekben hogyan helyezkednek el a tengelyek, az ablakok bal alsó sarkában lévő kis koordináta rendszerek mutatják. Ezek természetesen csak két-két irányt tudnak jelezni, mivel minden ablak két dimenziós. A pontos koordinátákat akkor tudhatjuk meg, ha kipipáljuk a **Display menü Coordinates** pontját. Ekkor az állapot sor jobb oldalán három számláló jelenik meg, amelyek a pointer X, Y és Z koordinátáit mutatják, ha az a három síknézeti ablak valamelyikében tartózkodik. Egyszerre csak két koordináta aktív amivel az egeret a képernyő mélységében nem tudjuk mozgatni. A koordináta rendszer origója most az ablakok közepére esik.

A perspektíva ablakban nincs koordináta számláló.

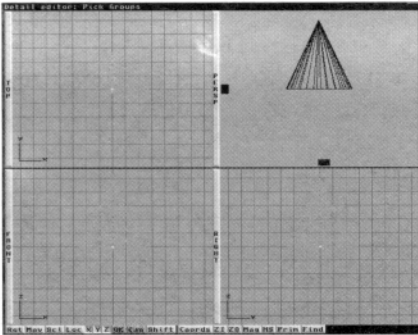
Ahhoz, hogy a megjelenítéssel való ismerkedésünket folytatni tudjuk, valamilyen tárgyat kell a képernyőre tennünk. Ennek mikéntjével csak később fogunk megismerkedni, ezért mindenféle magyarázat helyett mindenki tegye a következőket:

Functions/Add/Primitive

A megjelenő kérdésőben klikkeljünk a **Cone** kapcsolóra, mire újabb kérdésőt kapunk. Marad-

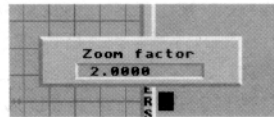


junk az előre felkínált alapbeállításoknál, válasszuk az OK-t. Máris készítettünk egy alapján álló kúpot, amely a program koordináta rendszerének origójában, a 0,0,0 pontban jelent meg. A kúp az ablakoknak csak kis részét tölti be, ahhoz hogy jobban megvizsgálhassuk, közelebb kell vinnünk hozzá a képzeletbeli nézőpontunkat. Válasszuk ki a **Display menü Zoom In** pontját és máris közelebbről szemlélhetjük művünket.



Ha újból messzebről akarjuk nézni a kúpot, használjuk a **Zoom Out**-ot. Ezt a kétfunkciót többször is alkalmazhatjuk egymás után, hogy egészen közelről, vagy ellenkezőleg, nagyon messziről nézzük rá a tárgyakra. Észrevehetjük,

hogy a **Zoom In** és **Out** egyformán hatással van mind a négy négyzetben. A nézőpont távolsága tehát nem függetleníthető az egyes ablakokban. Térjünk most vissza az eredeti távolsághoz. Válasszunk ki a **Display/Set Zoom**-ot. Egy kis input mezőt kapunk, amiben (ha jól állítottuk vissza a **Zoom**-ot) 1.0000 van írva. Írjuk át ezt az értéket 2-re. A régi számot letörölhetjük a **Del-gomb**bal, vagy a **Ctrl-X**-el, de az is elegendő, ha elébe írjuk az új értéket úgy, hogy a régi és új szám között maradjon egy szóköz.



Ha megvan, nyomjuk le az **Entert**-t. Ha mindent jól csináltunk, a kúpot újból olyan közelről szemlélhetjük, mint amikor egyszer használtuk a **Zoom In**-t.

Most írjuk a **Set Zoom**-ba 0.5-öt. Ennek hatására a kúp kétszer messzebről látszik, mint amikor létre-

Detail editor

hoztuk. A Set Zoom-mal tehát számszerűleg adhatjuk meg a nézőpont távolságát a tárgytól. Ez a távolság nem Imagine egységben van, hanem arányban. Az eredeti nézőpont az 1. a fele távolság 2. kétszer messzebb a 0.5. A 4-es érték azt jelenti, hogy az eredeti távolság 1/4-ről nézzük a tárgyakat.

A nézőpont távolságát nem változtathatjuk a végtelenségig. A minimális érték 0.0100, a maximális pedig 100.0000

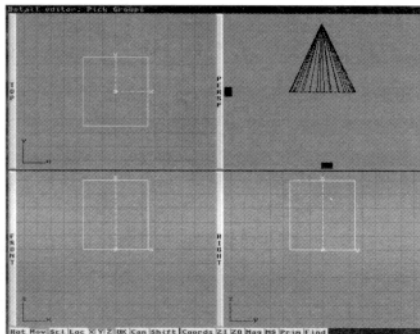
Állítsuk a Zoom faktort 1-re és hozzunk létre újabb kúpot, ugyan ezekkel az értékekkel. Látszólag nem történik semmi. A valóságban létrejött az új kúp, de ez is a világ tengelyrendszerének origójában, ugyan abban a méretben mint az előző, így csak egy látszik. Itt felmerült egy új kifejezés "**világ**". Természetesen nem a világegyetemről van szó, hanem a modellrendszerünkről. A program térbeli rendszrét úgy képzeljük el, mint egy világot, amelynek középpontjában a világ tengelyrendszerének origója van.

Tárgyak mozgatása

Nos vissza az előző problémához, hogyan választhatjuk szét a két kúpot? Erre több megoldás is kínálkozik, kezdjük a legegyszerűbb és leglátványosabbal. A kúp alapján, felülnézetben középen találunk egy kis sárga pontot. Vigyük oda a pointert és a hajszátkereszttel klikkeljünk erre a pontra. A tárgy színe narancssárgáról kék színűre vált. Erre a tárgyra azt mondjuk ki van választva. Ez azt jelenti, hogy a következő műveletek rajta hajtódhatnak végre. Most nyomjuk meg az **m** billentyűt önmagában, shift nélkül. Mindhárom síknézetben egy sárga négyzet jelenik meg a kúp helyett. A perspektíva ablakban

Tárgyak mozgatása

viszont továbbra is látható a kúp. Klikkeljük most a front nézetbe, majd a bal egér gomb folyamatos nyomva tartása mellett mozgassuk el az egeret az



ablakon belül. A sárga négyzet együtt fog mozogni a pointerrel, miközben a másik két síknézeti ablakban is megfelelően változik a pozíciója.

Ezt a sárga keretet **bounding box**-nak, magyarul befoglaló keretnek nevezzük. Feladata, hogy

interaktív transzformációknál meggyorsítsa a képernyőfrissítést, mivel ezt jóval gyorsabban ki lehet rajzolni, mint egy összetett tárgyat.

új fogalom: **Interaktív transzformáció**. Erre a mozgatás egy jó példa. Az interaktív transzformációk során a tárgyakat közvetlenül az egérrel tudjuk mozgatni, vagy forgatni, méretüket megváltoztatni. Az interaktivitáshoz tartozik, hogy a művelet eredményét folyamatosan szemmel kísérhetjük a monitoron.

Ha a pointer az ablakon kívülre ér, nem tudjuk tovább mozgatni a tárgyat. Ezen könnyen lehet segíteni, engedd el a bal gombot, vidd vissza a pointert az ablakba és "fogj újat" azaz ismételten nyomd meg a bal gombot. Így tovább lehet mozgatni a tárgyat. Az sem baj, ha a tárgy ér ki az ablak szélére, hisz a világ kiterjedése nem ér ott véget, ha a pointer benne marad az ablakban, a tárgy bármeddig elmozgatható. Próbáljuk ezt ki, mozgasd ki a sárga keretet az elől nézet jobb oldalán, annyira, hogy épp ne látszódjon. Ezután a tárgy az elől és felül nézetben már nem látható, de mivel a magas-

Detail editor

sági pozícióját nem változtathattuk meg, az oldalnézetben továbbra is látszik. Kikkelj most át ide és az előző módszerrel húzd lejjebb kissé a kúpot jelképező bounding boxot. Ha kész, nyomd le a space billentyűt. A keret helyén megjelenik a kék színű kúp, de hol a másik? Az az eredeti helyén, az origóban maradt, csak nem látszik, mivel azóta, hogy a másikat elhoztuk onnan, még nem volt teljes képernyőfrissítés. Aktiváld a **Display** menüből a **Redraw**-t és a másik kúp is megjelenik, narancssárga színben.

Idáig rendben, de az elől és felülnézetben nem látszik a második kúp, mivel kivittük az ablak szélén kívülre. Kikkelj bele ezen két nézet valamelyikébe majd nyomd meg a jobbra mutató kurzor nyilat. Az elől és a felülnézet balra elmozdult. Látszólag olyan, mintha a nézőpont ment volna jobbra, tehát arra, amerre mutató nyilat lenyomtad. Ha nem sokkal raktad ki a mozgató kúpot, akkor most az is láthatóvá válik minden nézetben.

Kikkelj most át a jobb nézetbe és nyomd meg újból a jobbra nyilat. A nézet képe balra mozdul, de az előlnézetben nincs változás, míg a felületnézet lefelé mozdul. Ebben semmi különös nincs, így természetes. Gondolj bele, a tárgyak a jobb nézeten Y irányban mozdultak, a felülnézetben ez lefelé van. Az előlnézet azért nem változik, mivel ott Y a mélység. A tárgy térben közelebb jött, de a párhuzamos vetítővonalak miatt a Z-X síkon ez nem észlelhető.

Akik ábrázoló geometriát vagy műszaki rajzot tanultak, tudom azoknak ez most furcsán hangzik, szerintük pont fordítva kell hogy legyen. Nos nekik is igazuk van, de a program amerikai, tehát az amerikai vetítési rendszert használja, nem az Európában megszokottat. Akinek nem lenne világos teljesen az egyes nézetek összefüggése, ajánlom mozgassa el ide-oda a kúpot és léptesse utána a nézőpontot.

Tárgyak mozgatása

Egy kis gyakorlás után biztos érthetővé válik az egész.

Az Imagine világnézetének megértése alapvető fontosságú, nélküle a háromdimenziós modellezés nem lehetséges.

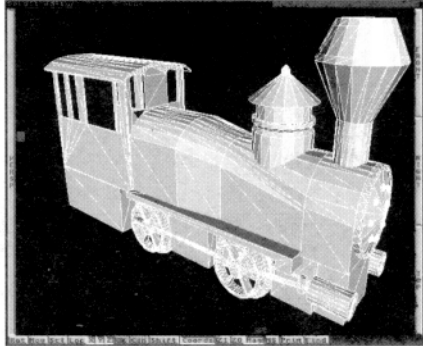
Idáig a perspektíva ablakban egyszerű drótvázás modelleket láthattunk, amelyekben az egyébként kitakart, nem látható vonalak is megjelentek. A **Display menü Solid** pontját kiválasztva a perspektíva ablakban már nem rajzolódnak ki ezek a vonalak. Ha ugyanitt a **Shaded**-et választjuk, négy nézetes rendben továbbra is kitakart vonalas, azaz "**hidden line**" perspektívikus a megjelenítést módja, de

ha ezt az ablakot teljes képernyősre kapcsoljuk, árnyalt fekete-fehér lapokból felépített tárgyat kapunk. Az alapbeállításra a **Wireframe** ponttal térhetünk vissza.

Ez a három opció, mint kitűnt az előzőekből, csak a perspektíva ablakra van hatással, a másik három nézetben mindig láthatjuk a kitakart éleket is, csakúgy mint egy drótvázás modellnél.

A Wireframe, Solid, Shaded sorrend a minőségi javulással párhuzamosan a szükséges számolási időt is növeli, lassúbb gépen nem célszerű folyamatos megjelenítésre használni a wireframe-en kívül mást.

Ha már nincs szükségünk a két kúpra, töröljük le őket. Ehhez előbb kikkelj a kúp alapján közepén lévő pontra, majd ha a tárgy kék színű lesz, használd a **Functions menü Delete** pontját. Ezt mindkét kúpon külön végezd el.



Detail editor

Az elmúlt néhány oldalon rengeteg új dolog, fogalom, művelet merült fel, megpróbálom ezeket elmagyarázni. Az egyik a tárgyak színének változása a szerkesztőben. Ennek a tárgyak állapotának megkülönböztetésében van jelentősége. Már láttuk, a **kék tárgy a kiválasztott**, azon hajtódnak végre a tárgymanipulációk. A **narancssárga az ún. kijelölt tárgy** színe. Ez annyiban különbözik a kiválasztottaktól, hogy még csak elő vannak készítve a manipulációkra. A **Pick/Select menü Pick Select** pontjának használata után ez a tárgy lesz a kiválasztott. Amikor egy új tárgyat töltünk be, vagy hozunk létre, az automatikusan kijelölt lesz.

A **lila** színt azok a tárgyak használják, amelyek egy időben vannak kijelölve és kiválasztva is. Ennek az eljárásnak az előnyére a munkánk során rá fogunk jönni.

A **fekete** a tárgyak alapszíne a szerkesztőben, ekkor sem kijelölve, sem kiválasztva nincsenek.

Hogyan jelölhetünk és választhatunk ki egy tárgyat? Erre az egyik módszert már megismertük, a tengelyének origójára kell kikkelni.

Tárgyak tengelye

No ismét egy új fogalom: **tárgy tengelye**. Az Imagine-ben minden tárgynak van egy saját tengelye. Ez független a világ tengelyétől, a **world axis**-től. A tárgy tengelyét egy hasonló lengelykereszt jelöli, mint a nézetekét, a különbség, hogy ez három irányú, és mindhárom irány meg van jelölve a kezdőbetűkkel.

Amikor egy tárgyat létrehozunk, a tengelye teljesen egybeesik a világ tengelyével, később ezt megváltoztathatjuk. Egy ilyen változtatás volt, hogy elmoz-

Tárgyak tengelye

gattuk a tárgyat. A tengelyeknek nem csak a pozíciója különbözhet, hanem az iránya is. Ez akkor következik be, ha a tárgyat elforgatjuk.

A tárgy tengelyének origóját egy sárga pont jelképezi. Ez a pont kitüntetett jelentőségű, mert nem csak hogy ezen keresztül lehet kijelölni a tárgyat, de a pozíciót is ezen keresztül lehet megadni. Tehát ha egy tárgy a 50,110,75 pozícióban van, akkor az azt jelenti, hogy a tengelypontja van ebben a pozícióban, függetlenül a tárgy térbeli elhelyezkedésétől. A tengely akár kívül is eshet a tárgyon, attól nagy távolságra is lehet.

A tárgy tengelyének van még egy fontos tulajdonsága, a tárgy forgatásakor forgástengelyül is szolgálhat. Ha a tárgy méretét változtatjuk, az is a tengelypont körül megy végbe. De ne vágjunk anyira a dolgok elébe, maradjunk a megkezdett témánál, a tárgyak kiválasztásánál.

Azt már megtanultuk, hogyan kell a tengelypontra kikkelve kiválasztani egy tárgyat. Hozzunk létre újból két kúpot, válasszuk szét őket, és azokon gyakoroljunk. Válasszuk ki az egyik kúpot, majd a másikat. Amikor a másodikat kiválasztjuk, az első kiválasztása megszűnik, színe visszavált feketére vagy narancsra. Mi a teendő, ha mindkettőt el szeretnénk mozgatni, de úgy, hogy egymáshoz viszonyított pozíciójuk megmaradjon? Egyszerre kell kiválasztani őket. Ehhez a **multi select** eljárást kell alkalmazni. Ne gondolj semmi bonyolultra, jelöld ki az első kúpot majd nyomd le a shift-et. Az állapot sorban megjelenik egy **Multi 1** felirat. Ez jelzi a multi módot, és azt, hogy egy tárgy van kiválasztva. A billentyű folyamatos nyomva tartása mellett válaszd ki a második kúpot, majd engedd el a shift-et. Így sikerült mindkettőt kiválasztani, együtt lehet őket mozgatni az m billentyű lenyomása után. Győződj

Detail editor

meg erről! A mozgatás most is a space-szel lehet véglegesíteni.

Mi van akkor, ha a mozgatás során rájössz, nem is ezt akartad, vissza az egész? Ha a space helyett az **Esc** billentyűt ütöd le, a mozgatás nem hajtódik végre, a tárgyak eredeti helyükre kerülnek vissza.

Több tárgy kiválasztása

Az iménti eljárással bármennyi tárgyat ki lehet egyszerre választani, de ha sokat kell, nem túl kényelmes. Erre az esetre van két gyorsabb módszer is. Menj a **Mode menü Pick Method** pontjára és ott válaszd a **Drag Box** alpontot. Ez egy kapcsoló, a **Click, Drag Box** és a **Lasso** közül egyszerre csak egy lehet, aktív. Az előző volt a Click. Drag Box-ban nem a tengelypontra kell klikkelni a kiválasztáshoz, hanem egy befoglaló keretet kell húzni és ha abba beleesik a tengelypont, a tárgy kiválasztódik. A befoglaló keretet úgy húzhatjuk meg, hogy a kurzort elvisszük mondjuk a bekeretezendő terület bal felső sarkába, majd az egér bal gombját folyamatosan nyomva tartva meghúzzuk a keret átlóját. A befoglaló keret elhelyezkedését folyamatosan mutatja a program. Ha bekerítettük a megfelelő területet, elengedhetjük a bal gombot, minek hatására létrejön a tárgyak kiválasztása. Több tárgy kiválasztásához most is használni kell a shiftet.

A harmadik kijelölő módszer, a **Laso**. Ekkor az egér bal gombjának folyamatos nyomva tartásával egy "lasszó"-t kerítünk a kiválasztandó tárgy vagy tárgyak tengelypontjai köré. Ahhoz hogy a tárgyak ténylegesen kiválasztódjanak, a lasszónak nem kell zártnak lenni. Amikor elengedjük a bal gombot, a lasszó két végpontját egy egyenessel összeköti a

Több tárgy kiválasztása

program, és az így kapott zárt területen belül lévő tengelypontokhoz tartozó tárgyak választódnak ki. Többes kiválasztáshoz most is kell a shift!

Az itt megismert kijelölő módszerek, nevesül a Click, Drag Box és Lasso, nem csak tárgyak kiválasztáshoz, hanem a későbbiek során megismert eljárásokban, a tárgyat alkotó pontok élek és felületek kiválasztásához is használhatók.

Méretváltoztatás, forgatás

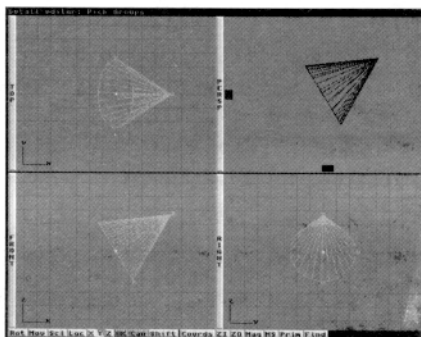
A mozgáson kívül van még két interaktív transzformáció: a **forgatás** és a **méretváltoztatás**. Próbáljuk ki először a forgatást.

Válaszd ki az egyik kúpot (most csak egyet) majd nyomd meg az "r" billentyűt. A kúp helyett most is a sárga keret jelenik meg. Vidd a pointert a felülnézetbe majd a bal gombot nyomva tartva mozgasd el az egeret az ablakon belül. A sárga négyzet a pointer mozgásával szinkronban fordul. A forgás tengelye egybeesik a kúp függőleges tengelyével. Forgasd el valamennyire a kúpot és nyomd le a space-t.

Változást természetesen alig látunk, mivel a kúpunk a magassági tengelyére, azaz Z-re forgásszimmetrikus. Mitől függ, mely tengely körül fordul el a tárgy? Nem mitől, hanem kitől. Tőlünk (is). Forgassuk meg most a másik kúpot. Miután megnyomtad az r-t és a tárgy sárga négyzetre cserélődött nyomd meg az y billentyűt. Vidd a kurzort az előlnézetbe, majd próbáld meg elforgatni a tárgyat. Érdekes módon az most az alapja azaz az Y tengelye körül fordul el. Miért van ez? Nos a tárgy egyszerre egy tengelye körül forghatható el, hogy ez melyik legyen, azt mi dönthetjük el. Erre az X,Y vagy

Detail editor

Z billentyűk szolgálnak. Hogy a tárgy melyik tengelye körül fog elfordulni, a képernyő alapján az user gadgetek közül az X, Y és Z feliratúak jelzik, úgy is kiválaszthatjuk a kívánt tengelyt, ha a megfelelő Usergadgetre kattintunk.



Idáig gondolom érthető, de biztos, hogy a tárgy a saját tengelye körül fordul el? Válaszd ki ismét az előző kúpot, azt amelyiket a Z körül már elforgattál. Forasd el a Z tengely körül annyira, hogy az X és Y tengelyei merőlegesen álljanak a világnak a nézet bal alsó sarkában lévő tengelyeivel. Mondhatnánk úgy is, hogy 90 fokot fordíts a tárgyon, annak eredeti állásához képest. Most nyomd le a space-t és győződj meg erről. Ha stimmel, nyomd le újból az r-t és próbáld meg forgatni a tárgyat az Y tengely körül. Gyanítom, nem az fog történni, amit vártál. A tárgy nem ez Y tengelye körül fog elfordulni, hanem a tárgy saját tengelyének origóján átmenő, a világ Y tengelyével párhuzamos egyenes körül. Minek akkor a tárgynak saját tengely, ha nem körülötte fordul el?

A világ tengelyrendszere magasabb prioritású, mint a tárgyak saját tengelye, de ez nem jelenti, hogy azt nem is lehet használni. Ha a kúpot annak Y tengelye körül akarod elforgatni, a forgatás megkezdése és a tengely kiválasztása után nyomd meg az I billentyűt, vagy klikkelj a "Loc" feliratú user gadgetre. Most a tárgy a saját, azaz lokális tengelye körül forgatható. Annak jelentőségét, hogy minden tárgy két tengelyrendszer hatása alatt áll, később érezni fogjuk.

Méretváltoztatás, forgatás

Gyakorlásképpen forgasd el a kúpokat kedved szerint, össze-vissza. Ha kellőképpen meguntad ezt a mókát, volna egy fogós ravasz kérdésem: Csak az alapja körül lehet a kúpot elforgatni? A válasz is az én dolgom lesz: Nem csak az alapja körül. Mindíg a tengelyrendszerének origója körül. Ezek szerint a tárgy tengelyének pozíciója megváltoztatható, lássuk hogyan.

A példa okáért próbáljuk meg az egyik kúpot a csúcsa körül elforgatni. Ehhez a tengelyt el kell mozgatni a csúcsra. Hogy a tárgyat hogy mozgat-hatjuk, már láttuk, most csak a tengelyével kell ezt tenni. Válaszd ki a kúpot és nyomd meg az **M**-et, de a **shift**tel együtt. A tárgy most is sárga keretre vált, de ha mozgatni próbálsz, csak a tengelye mozdul. Ha megfelel, nyomd le a space-t, majd az r a fogatáshoz. A kúp inmár a csúcsa körül fordul el.

Van egy haramadik interaktív transzformáció is, a méretváltoztatás. Töröld le a két kúpot, és hozz létre egy újat. Most legyen a feladat: Nyújtsd meg a kúpot, legyen sokkal magasabb. A méretváltoztatás az "s" billentyű lenyomásával kezdeményezhető, természetesen csak a kiválasztott tárgyon, vagy tárgyakon.

A méretváltoztatást szintén az egér bal gombjának lenyomva tartása melletti mozgatásával hajthatjuk végre. Próbáld is ki. Az eredmény: a kúp minden tengely mentén azonos arányban változtatja méretét. Mivel mi csak a magasságát akarjuk növelni, csak a Z tengely mentén kell engedélyezni a scale-t. Hogy ez a világ, vagy a tárgy saját Z tengelye, az most mindegy, mert e kettő épp egybeesik.

A transzformáció tengelyének kiválasztására most is az X,Y és Z billentyűk szolgálnak. Csak a forgatásra érvényes, hogy egyszerre csak egy tengely körül történhet, a scale esetében alpból be van kapcsolva az X,Y és Z is. Nyomd meg egyszer-egyszer

Detail editor

az X-et és Y-t, hogy csak a Z maradjon aktív. Ezt elérheted úgy is, ha a shift-Z nyomod le, ekkor ugyanis a másik két tengely automatikusan kikapcsolódik. Ez ugyanígy működik a többi tengellyel is.

Miután csak a Z tengely maradt aktív, tényleg a magassága változik a kúpnak. Megfigyelheted, hogy a méretváltozás is a tengelypont körül megy végbe. Tehát ha azt akarod, hogy a kúp csúcsa maradjon a helyén és lefelé nyúljon a tárgy, az előbb ismertetett módon told el a tengelyt a kúp csúcsába.

Foglaljuk össze, amit az interaktív tranzformációkról az előbbieken tanultunk:

A tárgyainkkal háromféle interaktív tranzformációt végezhetünk: mozgatást, forgatást és méretváltoztatást. Ezeket az m, r vagy s gombokkal kezdeméyezhetjük. Ha megfelel a tranzformáció, a space lenyomásával fogadhatjuk el. Az Esc billentyűvel anélkül léphetünk ki a műveletből, hogy elfogadnánk azt.

Egyik műveletből úgy is átléphetünk egy másikba, hogy közben nem fogadjuk el az előzőt a space-szel, hanem lenyomjuk a másik tranzformációnak megfelelő billentyűt. Például mozgatás közben lenyomjuk az r billentyűt, és máris forgathatjuk a tárgyat. A space ebben az esetben a teljes művelet-sort elfogadja, az Esc pedig az egész elveti.

Minden tranzformáció a tárgy tengelypontja mint középpont körül megy végbe. Azt, hogy a tárgy saját, vagy a világ közös tengelye legyen a viszonyítási alap, szintén megválaszthatjuk. Erre szolgál az l billentyű, vagy a "Loc" user gadget.

Azt is tudjuk szabályozni, hogy a műveletet mely tengely(ek) mentén kívánjuk végrehajtani. Forgatni egyszerre csak egy tengely körül lehet, ellenben mozgatni vagy méretezni két vagy három mentén is.

Tárgyak felépítése

Az egyes tengelyeket az X,Y és Z billentyűket lehet ki-be kapcsolni. Valamely tengelyt jelképező billentyű shiftelt használata a másik két tengelyt kikapcsolja.

A tárgy tengelye önállóan is manipulálható, mindhárom művelettel. Ennek jelentősége abban áll, hogy megváltoztathatjuk a tengelypontot, ezáltal a későbbi művelet középpontjának helyzetét, irányát, stb.

Amint idáig leírtam, kisebb eltérésekkel ugyan, de nem csak a Detail, hanem a Stage editorban is azonos módon használható.

A tárgyak felépítése

Miután megismerkedtünk az Imagine szerkesztőinek alapjaival, elmélyedhetünk a részletekben is. Ha már elkezdtük a Detail editort, maradjunk ennél. Mint korábban leírtam, a Detail az Imagine részletes tárgyszerkesztője. Többségében itt hozzuk létre tárgyainkat, itt látjuk el tulajdonságokkal, stb.

Ahhoz, hogy képesek legyünk tárgyakat létrehozni, meg kell ismerkednünk azok felépítésével. Az Imagine-ben többféle tárgytípussal találkozhatunk. Legnagyobb számban a síklapokból felépített térbeli alakzatok fordulnak elő. Ezeken kívül object-nek nevezzük még a matematikai formulával meghatározott gömböt, síkot, a B-Spline-okból felépített utakat, mozgáspályákat is, valamint a Stage editorban kezelt kamerát és fényforrásokat is.

Számunkra most a térbeli alakzatok, a testek a fontosak, a többivel később fogunk megismerkedni. Az Imagine-ben minden tárgy alapvető, ezért kötelező eleme a tengely. Tulajdonképpen a műveletek nagy részében, mint például a pozicionálás, mozgatás, forgatás, stb, ez reprezentálja a tárgyat. Minden

Detail editor

tárgynak csak egy tengelye lehet. A tengelyt a tengelykereszt és a tengelypont jelképezi. A tengelypont kitüntetett jelentőséggel bír.

A tárgyak (jelen esetben csak a testekre gondolok) mint már volt róla szó, síklapokból állnak. Ezek a síklapok a legegyszerűbb síkmértani elemek, háromszögek. Minden háromszöget három szakasz határol, ezeket éleknek nevezzük. Az élk mind-egyike meghatározható két végpontjának koordinátájával. Ezeket egyszerűen csak pontoknak nevezzük.

Ezzel el is jutottunk a tárgyak legkisebb építő elemeihez. Minden testet pontokból építünk fel, definiálva, hogy mely pontokat köti össze él, és mely élk között feszül felületelem. A pont, él és felületelem (háromszög) egymásra utaltsága egyirányú. Vagyis minden élhez tartozni kell két (és nem több) pontnak, de nem kell minden pontnak valamely élhez tartozni. Ez ugyanígy igaz az él-felületelem viszonyra is.

Egy pont több élnek is lehet végpontja, mint ahogy egy él is tartozhat több felületelemhez.

A tárgy alkotásában közvetlen jelentősége csak a felületelemeknek van, mivel csak ezeknek van síkbeli kiterjedése, a kiszámolt képen csak ezek láthatók. A pontokat és az éleket csak az editorokban láthatjuk, a dőtvasas modelljeinken.

Felvetődik egy fontos kérdés, jelelesül, ha minden tárgyat síklapokból állítunk össze, hogyan hozhatók létre görbék által határolt testek, mint például a gömb. Az ilyen tárgyakat sok, apró felületelemből kell összerakni, hogy minél jobban megközelítsük az elképzelt görbe felületet. Elméletileg végtelen számú síkelemet használva tökéletesen lefedhető a gömb felülete is, gyakorlatilag azonban ezt nem tehetjük meg, mivel a memória mérete és a gép sebessége mindig véges. Azt gondolom mindenki belátja, hogy

Primitív tárgyak

minél több elemből áll egy test, annál nagyobb lesz az azt leíró fájl mérete.

Gyakorlati munkánk során kell találni egy kompromisszumot amelyben az object fájl még megfelelő méretű, de a tárgy sem túl szögletes. Ezt a kompromisszumot egy matematikai eljárás is segíti, amellyel a program megpróbál két egymással érintkező, szöveget bezáró síkelem közé íves átmenetet generálni, ha kérjük. Ez az eljárás Phong vietnami matematikusról, az algoritmus megalkotójáról kapta a nevét.

Az elkészített tárgyat valahogy tárolni is kell. Az Imagine erre egy saját formátumot, a **TDD-t (Three Dimensional Data Description)** használja. Erről elég annyit tudni, hogy egy táblázatban tárolódnak a pontok koordinátái, valamint az, hogy mely pontokat köti össze él, és mely élek között van felület. A fájl tartalmazza a tárgy attribútumait is. Számmunkra a későbbiekben a pontok, élek és felületek sorrendje lesz fontos.

Primitív tárgyak

Ha minden tárgyat nekünk kellene pontokból felépíteni, az bizony időt rabló tevékenység lenne. Szerencsére a program rendelkezik előre definiált alapvető térbeli alakzatokkal, amelyeket felhasználhatunk bonyolultabb tárgyaink felépítéséhez. Ezeket az egyszerű tárgyakat primitíveknek nevezzük. Egyik fajtájával, a kúppal már találkoztunk a könyv egy korábbi szakaszában. Most nézzük át az egyes primitíveket részletesen is.

Primitívet létrehozni a **Functions menü Add** pontjának **Primitive** alpontjával lehet. Ezt kiválasztva egy kérdező jelenik meg, ahol kiválaszthatjuk a megfelelő primitív típust.

Detail editor

Hat féle tárgy közül választhatunk, ezek:

Sphere - gömb

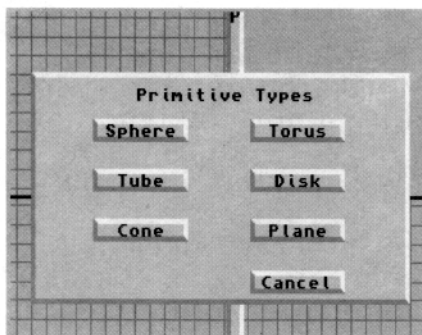
Tube - cső, vagy henger

Cone - kúp

Torus - kör keresztmetszetű gyűrű

Disk - korong

Plane - síklap



A **Cancel** nem egy tárgytípus, ezzel a kapcsolóval léphetünk ki a kérdezőből, anélkül hogy primitívet hoznának létre.

Ha valamely primitív nevét viselő kapcsolóra klikkelünk, újabb kérdezőt kapunk, ahol az adott tárgy paramétereit állíthatjuk be. Nézzük sorba az egyes tárgyakat és a beállítási lehetőségeket.

Sphere

Három input mezőt találunk a **Sphere Parameters** kérdezőben. Az első a **Radius**, ahol a gömb sugarát kell megadni, Imagine egységben. A második a **Circle Sections**, amely a gömböt alkotó "gerezd"-ek száma. A **Vertical Sections** pedig a vízszintes szeletek száma.

Minél több gerezdből és vízszintes szeletből építjük fel a gömböt, annál simábbnak tűnik, de a fájl méret ezzel együtt drasztikusan emelkedik. Hogy hány pontból álljon a tárgy, attól is függ, milyen messziről fog látszódni a képen. Minél messzebbi lesz, annál kevesebb pont is elegendő az elkészítéséhez.

Az input mezők alatt található egy kapcsoló **Stagger Points** néven. Ezt bekapcsolva a program a tárgy kiszámolásakor minden két pont közé generál egyet, tovább finomítva a tárgyat. Ez a Stagger point

Primitív tárgyak

nem növeli a fájl méretét és nem okoz jelentős számítási idő növekedést, sőt ellenkezőleg, kisebb lehet vele a fájl, mivel kevesebb pont is elég a felépítéséhez.

Az **OK** és a **Cancel** gondolom nem szorul magyarázatra, a későbbiek során sem foglalkozunk velük.

Miután beállítottuk a nekünk megfelelő paramétereket és elfogadtuk az **OK**-val, a világ koordináta-rendszerének origójában megjelenik a gömbünk, amelynek tengelypontja pontosan a közepébe esik. Itt jegyzem meg, hogy minden primitív, sőt minden egyéb tárgy létrehozásakor a 0,0,0 pozícióba kerül.

Miután megszemléltük újonnan létrehozott gömbünket töröljük le, hogy ne zavarjon bennünket a következő tárgy létrehozásában. A törlés menete egyszerű, ki kell választani a tárgyat, majd alkalmazni a **Functions** menü **Delete** pontját. Ezzel egyenértékű a jobb **Amiga** és a **d** billentyű egyidejű lenyomása.

Vissza a **Primitive Types** kérdezőbe, újabb primitívet hozunk létre. Aki nem szereti a menük használatát, az egyszerűsítheti a kérdező előhívását a "**Prim**" gadgettel.

A következő tárgyunk a **Tube**, azaz cső. Itt a következő paramétereket adhatjuk meg:

Radius - A cső keresztmetszetének sugara.

Height - A cső hossza.

Circle Sections - A csövünk palástját ennyi részből állítjuk össze, azaz a kerületet ennyi egyenes határolja.

Vertical Sections - Ennyi szeletből áll a cső hosszában.

Az **input** mezőkön kívül három kétállású kapcsoló található még a kérdezőben. Az első, a **Stagger Point** már ismerős, feladata ugyanaz, mint a gömbnél.

Detail editor

Close Bottom - Alaphelyzetben a csövünk mindkét vége nyitott, mint minden rendes csőnek. Ezzel a kapcsolóval bezárhatjuk a cső alsó nyílását.

Close Top - Mint előbb, de a cső tetejét zárja be.

Hiába zárjuk be a cső mindkét végét, az nem lesz tömör. Általánosságban elmondható, hogy a primitívek nem tömörek, sőt anyagvastagságuk sincs. Ez nyilvánvaló, ha belegondolsz, hogy nulla vastagságú síklapokból állnak a tárgyak. A ray-tracing gyakorlatában ennek nem sok jelentősége van.

Ha kigyönyörködted magad a csőben jöhet a következő tárgy a **Cone**, azaz a kúp. Ezzel már találkoztunk, de akkor nem részleteztük.

Radius - A kúp alapjának kerülete.

Height - A kúp magassága.

Circle Selections - A kúp palástját alkotó síklapok száma.

Vertical Sections - A kúpot alkotó vízszintes szeletek száma.

A kúp alapjának zártta tételére a **Close Bottom** kapcsoló szolgál.

A következő primitív amivel megismerkedtünk a Torus fedőnevű, kör keresztmetszetű gyűrű. Úgy képzeljük el, mint a karikára hajtogatott csövet, pld bicikli tömlőt. Ennek a következő paramétereit adhatjuk meg:

Ring Radius - A gyűrű középvonalának sugara.

Tube Radius - A gyűrűt alkotó, karikába hajtott cső sugara.

Ring Sections - A gyűrű kerülete- azaz a karikára hajtott cső hossza- ennyi szeletből áll.

Tube sections - A cső keresztmetszetét alkotó egyenesek száma.

A **Stagger Points** jelentése a szokásos.

Primitív tárgyak

Az eddig tárgyalt négy primitív mind térbeli alakzat volt. A most következő kettő síkidom.

Sorban a következő a **Disk**, magyarul korong. Ennek mindössze két paramétert kell megadni.

Radius - Értelemszerűen a korong sugara.

Sections - A korong területét ennyi szekció határozza meg, azaz ennyi pont övezi a területet.

Hengeres, kör keresztmetszetű tárgyknál a területet alkotó vonalak száma természetesen azonos az azt alkotó pontok számával. Nem így van ez az utolsó primitívvel, a **Plane** nevű síklappal. Ennek négy paramétere van.

Width - Szélesség, Imagine egységben.

Height - Magasság, Imagine egységben.

Horizontal Sections - A vízszintes szeletek száma ennél egyel több pont alkotja a tárgy szélességét

Vertical Sections - Mint az előző, de függőlegesen

Nos ennyi primitívet hozhatunk létre az Imagine-ben. Ezek mindegyikének vannak alap paraméterei, amelyek az input mezőkben megváltoztathatók. A return használata fontos egy-egy változtatás lezárásához, nélküle az érvénytelen lesz.

Az élesszeműeknek biztosan feltűnt, hogy a **Functions/Add** menüpontnak a **Primitive**-en kívül van még néhány alpontja. Ezek mindegyike valamilyen tárgyat ad a szerkesztőbe. Ezek között az alpontok között is van egy **Spere**. Ez azonban nem a szokványos felépítésű, nem síklapokból áll, hanem egy matematikai egyenlettel leírt görbékéből. Ez a gömb egy "tökéletes gömb" de előnyei mellett (gyors leképezés, kisebb fájlméret, stb) vannak hátrányai is. Például nem lehet eltorzítani, mindig gömb alakú marad a tárgy, nincsenek pontjai, amelyekhez hozzá lehetne férni, stb.

Detail editor

A **Ground** egy X és Y irányban végtelen méretű, Z irányban 0 kiterjedésű sík. Ez is egyenletek által van meghatározva, pontjai, élei, felületelemei, nincsenek. Az **Imagine**-ben a talaj modellezésére használjuk.

A **Functions/Add** többi alpontjával később fogunk megismerkedni, mivel megértésükhöz olyan ismeretek is szükségesek, melyekről idáig nem volt szó.

Összefűzött tárgyak

Az imént ismertetett tárgyak az **Imagine** modellező rendszerének csak egy részét jelentik, bár ezek felhasználásával jóval bonyolultabb objecteket is elkészíthetők. Hogyan? Vegyünk egy egyszerű példát, készítsünk egy súlyzót, amolyan rajzfilmszerűt. Ez egy rúdból áll, mindkét végén egy-egy gömbbel. Tervezzük meg először gondolatban a munkát. Szükségünk van két gömbre, amit egymástól eltávolítunk, és közéjük rakunk egy kis átmérőjű hengert.

Ha kész a terv, lépünk a tettek mezejére, először készítsük el a két gömböt, az alap paraméterekkel. Válaszd ki az egyiket, majd az X tengely mentén mozgasd el jobbra, mondjuk 300 egységgel. Jusson eszedbe, hogyan lehet a mozgatót lekorlátozni csak az X irányba!

Hogy 300 egységnyi legyen a mozgató, lépj be a front nézetbe és használd ki a rács nyújtotta előnyöket. Ahhoz, hogy ki tudjuk számítani a 300 egységet, ellenőrizd le a rácsméretet, a már megismert **Display/Grid Size** menüponttal. Hagyjuk ezt most az alapértelmezés szerinti 20 egységen. A 300 egység tehát 15 rácspontnyi távolságot fog jelenteni, ennyivel húzd el az egyik gömböt.

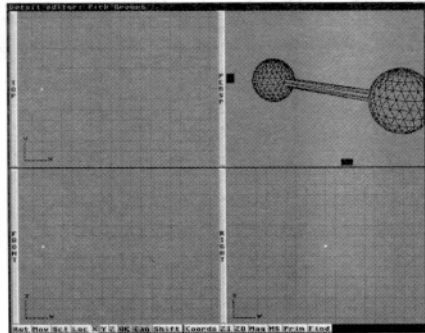
Összefűzött tárgyak

Végső soron nem annyira fontos a nagy pontosság. A későbbiek során majd megismerkedünk olyan eljárással, amivel precíz mozgásokat, pozicionálásokat és más transzformációkat hajthatunk végre.

Tehát megvan a két súly, már csak rudat kell közéjük rakni. Erre a célra tökéletesen megfelel a Tube primitív, még a végeit sem kell lezárni. A cső magassága lehet 300 egység, legfeljebb beér a gömbök közepéig. Az átmérőnek kicsinek kell lenni a hosszhoz képest, mint a valóságban. Számszerűleg ez 8-10 egységet jelent. Mivel az átmérőt ilyen kicsire vettük, elegendő jóval kevesebb Circle Section, azaz körszelet is, mondjuk 8.

A cső a frontnézetben függőlegesen áll, derékszögben a szükséges pozícióra. Mivel a tengelypontja, amely az alsó végén van, egybeesik a nem mozgatótt gömbével, nincs más teendőnk, mint az Y tengely körül elfogatni a megfelelő pozícióba.

Ha ez megvan, majdnem kész a súlyzó. Most jön azonban egy újabb probléma, hogyan lesz ebből a három tárgyból egy? Ugy, hogy egybefűzzük őket. A nyél még ki van jelölve, a shift használatával válaszd hozzá a két gömböt is. Most használd a **Functions menüből a Join** pontot. A három tárgy összefűződik egygé. A tengelye azé a tárgyé lesz amelyiket elsőnek választottál ki. Most közelíts rá a Zoom-mal annyira, hogy a perspektíva nézetet teljesen kitöltse a súlyzó. A program **Quick Render** funkciójának segítségével elkészítjük a tárgy képét. Használd a **Project menüből a Quick Render-t**. A megjelenő



Detail editor

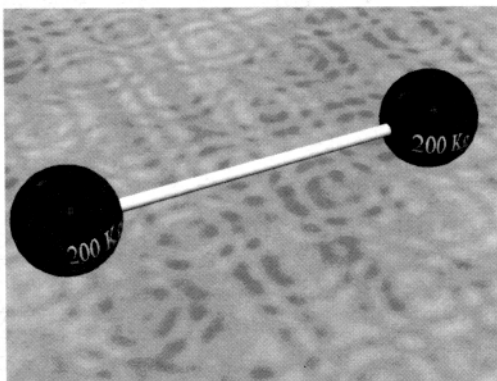
kérdőben megadhatjuk az ideiglenes fényforrás vízszintes és függőleges fénysugárzását, ennek azonban gyakorlati jelentősége nem sok van, hagyjuk alapértékeken.

A kérdésben van egy kétállású kapcsoló, "**Don t Add Light Source**" néven. Ezt bekapcsolva nem adódik ideiglenes fényforrás a szerkesztőhöz. Akkor van jelentősége, ha valamelyik tárgyunk rendelkezik fényforrásszerű tulajdonságokkal és annak a fényével akarjuk a képet elkészíteni. Mivel most ez az eset nem áll fenn, hagyjuk ezt a lehetőséget.

Géptől függően rövidebb-hosszabb idő után elkészül a gyorsfotó, amely alap esetben a **RAM**-ban foglal helyet, **Quickrender** néven. A képet a program meg is mutatja nekünk. Ha kigyönyörködted magad az első munkánkban, folytassuk.

Nyomd meg az egér bal gombját, vagy az Esc billentyűt, hogy eltüntesd a gyorsfotót. Most a program megkérdezi, letörölje-e a képfájlt. Ha van még vele valamilyen célod, válaszd a **No**-t, egyébként a **Yes**-t.

Bár ez a súlyzó hagy még kívánni valót maga után, érdemes lenne kimenteni a tárgyat, ha másért nem, hogy gyakoroljuk a műveletet. Gondolom a többség rájött már, hogy erre az **Object menü Save** pontja szolgál. A kimentéshez a tárgyat ki kell választani. A megjelenő fájl szelektor a szokásos. A névválasztásra csak az alkalmazott géptípus fájlrendszere szolgál megkötéssel



Összefűzött tárgyak

Javaslom a megkülönböztetésre használj **.iob** kiterjesztést a fájlnevében.

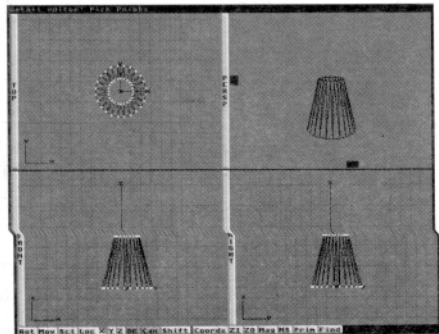
Nem árulok el titkot, ha elmondom, hogy a tárgyat betölteni ugyanennek a menünek a **Load** pontjával lehet. A betöltött tárgy mindig a kimentéskori pozícióba és irányba kerül.

Az előző példa igencsak egyszerű volt, lássuk valami bonyolultabbat, készítsük egy keljfeljancsit. Először most is átgondoljuk a feladatot. Szükségünk lesz egy csonka kúpra és két félgömbre, melyből az egyik a kúp aljához, másik a tetejéhez illeszkedik. Csonkakúp, félgömb, ilyen primitívek nincsennek is! Majd csinálunk. Csonka kúpot kúpból, félgömböt gömbből!

Gyerünk sorba, készítsük el előbb a kúpot. Hozz létre egy Cone primitívet, amely alapjának sugara 50 egység, magassága 200 egység, 24 körszeletből és 2 vízszintes szekcióból áll. Ebből a kúpból könnyen készíthetünk csonka kúpot, ha levágjuk a felső vízszintes szekcióját. Ezt végrehajthatjuk úgy is, ha töröljük az ezt a részt alkotó felületelemeket, vagy az éleket. Az is megoldás, ha töröljük a csúcspontot, mivel ez a fölös rész éleinek közös végpontja, nélküle ezek az élek, következésképp a felületelemek sem létezhetnek.

Utóbbi módszerhez a következőket kell tenni:

Válaszd ki a kúpot, majd a **Mode** menüből kapcsold be a **Pick Points** módot. Láthatóvá válnak a tárgy pontjai, ezeket tudjuk kiválasztani. Az eljárás ugyan az, mint a tárgyak esetében, itt is lehet használni a



Detail editor

Click, Drag Box és **Lasso** kijelölő eljárásokat. Mivel csak egy pontot kell kiválasztani, maradhatunk a **Click**-nél. A csúcspontra klikkelve az narancs színű lesz.

Most töröld le a **Delete** funkcióval. Ezzel kész a csonka kúp, lépj vissza **Pick Group** módba.

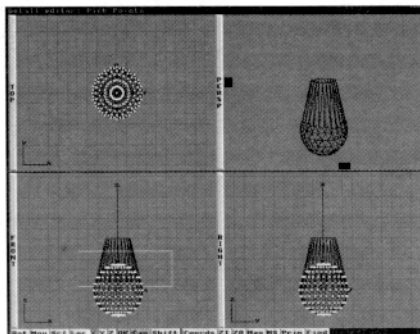
Hozzuk létre az alsó félgömböt. Erre a célra tökéletesen megfelel egy alap paraméterekkel rendelkező gömb. A gömb pont jó helyen jön létre (feltéve, ha a kúpot nem mozdítottad el), válaszd is ki. Vissza **Pick Points**-ba.

Láthatod, hogy csak a gömb pontjaihoz lehet most hozzáférni. Egyszerre csak egy tárgy pontjai manipulálhatók. Ha több tárgy is ki volt választva, amikor **Pick Points**-ba belépünk, annak a pontjaihoz tudunk hozzáférni, amelyik hamarabb lett kiválasztva.

Hogy ne zavarjon a kúp, használd a **Redraw**-ot, aminek hatására eltűnik a képe (a tárgy nem törlődik!). **Pick Points** módban, amikor képernyőfrissítést hajt végre a program, csak a kiválasztott tárgyat rajzolja fel. Van amikor ez hátrányos, például, ha a tárgy pontjait egy másik tárgyhoz kell igazítani. Ekkor úgy lépj be **Pick Points**-ba, hogy a manipulálandó tárgy után kiválasztod a másikat is. Ekkor a másik kiválasztott tárgyat is újrajzolja frissítéskor, bár ennek pontjaihoz nem férhetünk hozzá.

Az előbb leírtak érvényesek a később említésre kerülő **Pick Faces** és **Pick Edges** módokra is.

E kis kitérő után vissza az eredeti feladathoz,



Összefűzött tárgyak

félgömböt kell készíteni a gömbből. Remélem az előző gyakorlat után ez már nem okoz gondot. Vigyázz, a gömb egyenlítőjén lévő pontokat ne töröld le, mert akkor kevesebb marad a tárgyból, mint fele!

Hátra van még a felső félgömb. Ennek sugara legyen 25 egység. A többi paraméter marad alapon.

Ez a harmadik tárgy, amelyik a 0,0,0 pozícióban van, nehéz köztük a választás. Az Imagine többféle kijelölő-kiválasztó eljárást kínál a probléma megoldására. Ezeket a **Pick/Select** menün keresztül érhetjük el. A **Pick All** ponttal a szerkesztőben lévő összes tárgyat egyszerre kiválaszthatjuk.

Amikor tárgyakat hozunk létre, azok létrehozásuk sorrendjében egy listába kerülnek. A sorrendet később szabadon megváltoztathatjuk. A **Home** menüpont ennek a listának az élén álló tárgyat jelöli ki. A **Select Next** és a **Select Prev** a jelenleg kijelölt utáni, illetve előtti tárgyat jelöli ki a listából. Ha a tárgy nem látszana, akkor a szerkesztő ablakok úgy mozdulnak el, hogy az a közepére kerüljön.

A **Recenter**-rel bármelyik tárgyat a szerkesztők közepére állíthatjuk. A menüpont aktiválása után a középre állítandó tárgy tengelyére kell klikkelni.

Hogy mi az értelme a kijelölésnek, amikor csak a kiválasztott tárgyakon lehet műveletet végrehajtani? A következő menüpont, a **Pick Select** választ ad erre. Hatására az éppen kijelölt tárgy kiválasztottá válik, művelet alá lehet vonni. Ezek nagyon hasznos funkciók, nagyon megkönnyíti a munkánkat, ha a hozzájuk tartozó billentyűzet-rövidítéseket, amelyek a jobb oldali Amiga (PC-n Alt) billentyű és egy betű kombinációi, megtanuljuk. A **Pick Select**-nek nincs ilyen rövidítése, de ez meg hozzá van rendelve a Preferences-en keresztül az **F1**-hez.

Az **Unpick Select** a kijelölt tárgy kiválasztottságát szünteti meg, az **Unpick Last** pedig az utolsónak

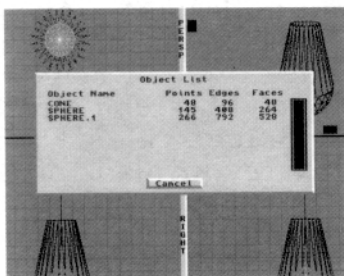
Detail editor

kiválasztottét. Utóbbi funkció főleg többes kiválasztáskor hasznos, például ha olyan tárgyat választottunk ki, amelyiket nem akartunk.

Az előbb említettem, hogy a tárgyak listabeli sorrendjét meg tudjuk változtatni. Ez a tárgyra semmilyen hatással nincs, nem változnak meg paraméterei, stb. Ha sok tárgy van egyszerre a Detailban, érdemes a sűrűbben használtakat a lista elejére rendezni, mivel így a **Home**-mal és a **Select Next**-el könnyű elérni azokat.

A rendezés menete igen egyszerű, sorban ki kell választani a tárgyakat többes kiválasztással, majd használni a **Short**-ot. A tárgyak új sorrendje azonos lesz a kiválasztásuk sorrendjével. A rendezéskor ki nem választott tárgyak, egymáshoz viszonyított sorrendjüket megtartva, a lista végére kerülnek.

Ha tudjuk a tárgy nevét, a **Find by Name** kérdezőbe beírva azt, közvetlenül is kijelölhetjük. Legtöbbször nem tudjuk a nevét, mint ahogy most sem, ezen segít a **Find Requester**. Ez a menüpont egy kérdezőt hoz elő, amiben fel vannak sorolva a tárgyak a már említett lista sorrendjében. A nevek után láthatjuk, hogy az adott tárgy hány pontból, élből és felületelemből (háromszögből) áll. Ha a sok tárgy nem férne el egyszerre a listában, a jobb oldali gördítőléccel lehet lapozni.

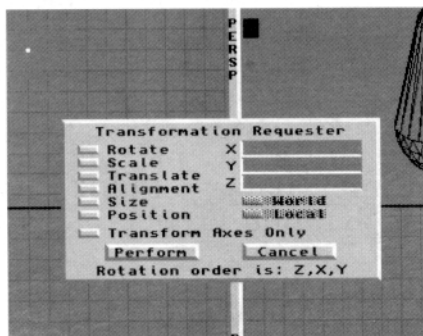


A **Pick/Select** menü többi elemével később foglalkozunk, egyelőre elég ennyi a továbbhaladáshoz. Térjünk vissza a keljfeljancsihoz.

Miután az új módszerek ismeretében kiválasztottuk a kisebb gömböt, Z pozitív irányában el kell tolni pontosan 100 egységnyit. Itt már lényeges a pontosság, ezért ismerkedjünk meg egy új funkcióval.

Pontos transzformációk

Pontos transzformációk



Válaszd ki a kisebb gömböt, majd használd az Objects menü **Transformation** pontját. Egy kérdező jelenik meg, amellyel transzformációkat hajthatunk végre a kiválasztott tárgyakon. Ezek a transzformációk hasonlóak, mint az interaktívak, de itt nem az

egérrel, hanem pontos számértékekkel irányítjuk a műveleteket.

A transzformáció típusát a bal oldali listából választhatjuk ki, a paramétereket, a három tengely szerint külön, a jobb oldali három input mezőben adhatjuk meg.

Bizonyos műveletnél választhatunk, hogy a világ (**World**) vagy a tárgy saját (**Local**) koordináta rendszere szerint kívánjuk a végrehajtást. Ezt az input mezők alatti két kapcsolóval lehet kiválasztani. Ahol nincs választási lehetőség, ott az egyik kapcsoló inaktív.

A "**Transform Axes Only**" kapcsolót kiikszelve a kiválasztott művelet csak a tárgy tengelyén hajtódik végre.

Az egyes műveletek a következő három típusok valamelyikébe tartoznak: Forgatás, helymeghatározás és méretezés.

Az első típusba tartozik a **Rotate**, amivel a három tengely közül a szögfokokban megadott mértékben forgathatjuk el a tárgyakat. Szintén a forgatás csoportjába tartozik az **Alignment**. Itt a tárgy és a

Detail editor

világ tengelyének egymással bezárt szögeit adhatjuk meg fokokban. A forgatás előbb az X, majd az Y, végül a Z tengelyek körül megy végbe. Ennek akkor van jelentősége, ha egyszerre több tengely körül forgatunk.

A második művelettípusba, a helymeghatározásba szintén két művelet tartozik. A **Translate** a tengelyek mentén Imagine egységben megadott mértékben mozgatja el a tárgyakat. Ennek párja a **Position**, amely a tárgyat a világ tengelyrendszerének megadott koordinátájába mozgatja.

A harmadik típusba tartozik a **Scale**, amely a megadott arányban változtatja a tárgy méretét. A 2-es érték duplázza a tárgy kiterjedését az adott tengely mentén, a 0.5 felezi. Párja a **Size** abszolút méretet határoz meg.

Mint látható, mindhárom típusba két művelet tartozik, egy relatív és egy abszolút. Egyszerre három transzformációt hajthatunk végre a tárgyakon, de mindhárom típusból csak egyet. Minden műveletre külön-külön megadható, hogy melyik tengelyrendszer legyen a viszonyítási alap (már ahol lehet választani). A Transformation nem csak tárgyakra alkalmazható, hanem az azt alkotó pontokra, élekre, vagy felületekre is. Ezen elemek forgatásakor, méretezésekör megjelenik a kérdezőben egy új kapcsoló, az **Auto Centered**, amely alap esetben be van kapcsolva. Ez azt jelenti, hogy a pontok (élek, felületek) nem a tárgy tengelye, hanem a kiválasztott elemek mértani középpontja körül hajtják végre a műveletet.

A **Perform**-ra klikkelve hajtódik végre a transzformáció.

Nekünk most a Translate funkcióra van szükségünk, a világ Z tengelye mentén + 100 egységgel kell elmozgatni a kis gömböt, hogy a csonka

Pontos transzformációk

csonka kúp tetejére helyeződjön át. Természetesen használhatod a **Position**-t is, hogy a tárgy a 0,0,100 koordinátába kerüljön.

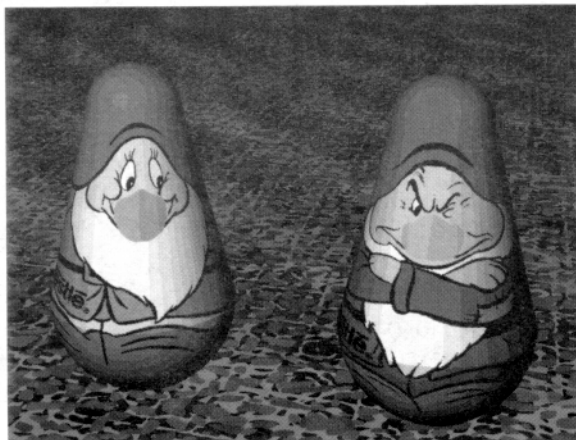
Hogy miért pont ide, azt most nem magyarázom meg, legyen elég annyi, hogy csak.

Ha megvan, töröld le a gömb felesleges pontjait. Ezzel kész a keljfeljancsink, csak össze kell fűzni. A változatosság kedvéért most ne a Join-nal fűzzük egybe a tárgyainkat, hanem a **Merge**-dzzsel. Ez is a **Functions** menüben található, közvetlenül a Join alatt. Annyiban különbözik a Merge a Join-tól, hogy ez kiszűri a kettős pontokat, éleket és felületeket, optimalizálva a fájl méretét. Ha sok elem van átfedésben, igen hasznos. Mivel nem egyszerűen összekapcsolódnak a tárgyak, a végrehajtási ideje kissé hosszabb mint a Join-é.

Állítsd be ezt a tárgyat is a perspektíva ablakban, és a Quick Renderrel készítsd el a képét.

Az idáig megismert eljárásokkal már igen sokféle tárgyat el tudunk készíteni, de még koránt sem értünk a Detail

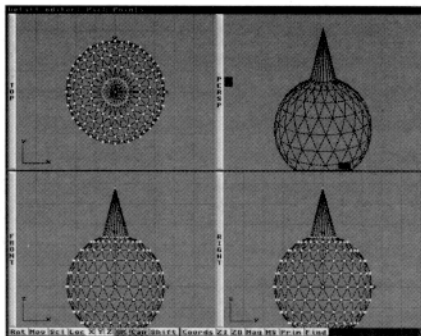
editor lehetőségeinek végére. Készítsünk most egy karácsonyfadísz üveggömböt. A kiindulási alap természetesen a gömb primitív.



Detail editor

Alkotó pontok átrendezése

Válaszd ki és menj át Pick Point módba. A tervem a következő: a gömb tetején lévő pontot kihúzzuk, elkészítve az üveggömb szarát. Jelöld ki a legfelső pontot. Most ismét egy kérdés: Hogyan fog elmozdulni innen? Talán az interaktív mozgatót ha megpróbálnánk. Nyomd meg az m billentyűt. A kijelölt pont és a belőle induló élek sárga színűek lesznek, a szokott módon mozgathatod azokat. Húzd fel annyira, mint a képen látszik.



Ha kész, zárd le a műveletet a space-szel. Ez még nem az igazi, le kell tompítani a csücsöt. Hogyan? Ez fogós ravasz kérdés, ha ugyanis újabb pontsort húzunk fel, túl széles lesz a szár, ha pedig több pontból hozzuk létre a gömböt, túl nagy lesz az object fájl. A legjobb volna csak a gömb tetején sűríteni a pontokat. Talán ha a kihúzott éleket megfeleznénk, újabb pontokat hozva létre.

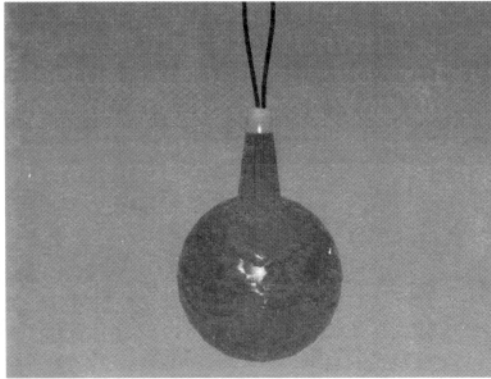
Ez egy jó ötlet (nem hiába, az enyém), sőt, még van is ilyen funkciója a programnak. Ehhez csak annyit kell tenni, hogy kijelöljük az elfelezendő éleket. újabb probléma, de nem fog ki rajtunk, a Mode menüből válaszd ki a **Pick Edges**-t. Ebben a módban a tárgyakat alkotó éleket tudjuk kiválasztani és manipulálni. A kiválasztást úgy lehet végrehajtani, hogy **Click Pick Method**-ban mindkét végre ráklikkelünk, vagy a **Drag Box** és **Lasso** módok valamelyikében mindkét végpontját bekerítjük. Több él egyszerre

Új tárgy létrehozása

történő kiválasztásához a shift-et kell használni.

Nekünk most a **Drag Box** felel meg legjobban, kerítsd be vele a kihúzott élek végpontjait, miközben lenyomva tartod a shiftet. Az élek felezését a **Functions** menü **Fracture** pontjával lehet végrehajtani.

Ha megvannak az új pontok, térj vissza Pick Points-ba és válaszd ki őket, majd mozgasd el a Z tengely mentén felfelé, az első ponttal egy magasságba. Most már jobban néz ki a karácsonyfadíszünk.



Új tárgy létrehozása

Eddig csak a primitívek továbbalakításával hoztunk létre tárgyakat, ideje volna valami újat is alkotni. Mivel minden tárgy alapja a tengely, új tárgy létrehozását ennek ekészítésével kell kezdeni. Tehát **Funkctions/Add/Axis**. A 0,0,0 pontban megjelenik egy tengely. Válaszd ki, majd Front nézetbe átmenve válaszd ki a **Mode** menüből az **Add Points** módot.

Legyen az első tárgy, amit felépítünk, egy háromszögű vonalzó. A vonalzó éleinek illik egyenesnek

Detail editor

lenni, és az is jó, ha két éle derékszöget zár be.

Ezt két módon érhetjük el. Az egyik, hogy minden pontját a Transformation kérdezőjén keresztül numerikusan pozicionáljuk, de ez a macerásabb. A másik lehetőség, hogy kihasználjuk a rács nyújtotta előnyöket. Hiába próbáljuk meg azonban kézzel a rácsra tenni a pontokat, még ez sem tökéletes.

Szerencsére a program rendelkezik egy olyan funkcióval, amely a letett pontokat pontosan a rács keresztetződéseire igazítja. Bekapcsolni a **Mode menü Pick Method** pontjának **Lock** alpontját kipipálva lehet. Így már nincs akadály a vonalzó elkészítésének.

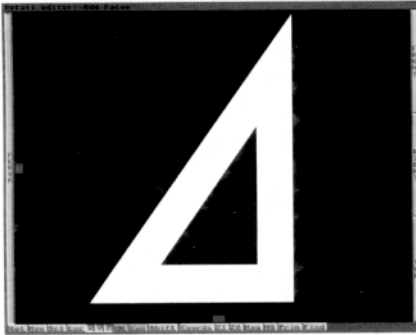
Először a külső körvonalának három csúcspontját határozd meg úgy, hogy az előlnézetben a megfelelő helyekre klikkelve lerakod a pontokat. Ezután lépj át **Add Edges** módba, hogy csúcspontokat élekkel köthessed össze. Ez úgy történik, hogy egymás után a leendő él két végpontjára klikkelsz.

Ha kész a körvonal, határozzuk meg a belső nyílást. Alkalmazhatjuk az előző módszert is, de szerintem lassú és nehézkes, hogy külön határozzuk meg a pontokat, majd az éleket. Ehelyett inkább válaszd ki az **Add Lines** módot. Ez ötvözi az Add Points-ot és az Add Edges-t. A klikkeléssel most is pontokat rakunk le, amelyeket a Lock a rácsra igazít, de a pontok rögtön éllel kötődnek össze.

A vonalzó körvonala már kész, de a **Quick Render** nem mutat semmit. Ez teljesen természetes, gondolj csak arra, amit a könyv egy korábbi részében írtam, kiszámított képen csak a felületelemek látszanak, a pontok és élek nem. Nincs más hátra, felületeket kell meghatározni a tárgyon.

Lépj át **Add Faces** módba, ahol végrehajthatjuk a feladatot. A felületelemek, amelyek mindig háromszögek, hasonló módon hozhatók létre, mint az élek

Kinyomás térben



az Add Edges-ben. Egymás után ki kell választani a három végpontot. Amint megvan a harmadik csúcsa is a háromszögnek, a program létrehozza a háromszögű felületelemet.

A határoló éleknek nem, de a csúcspontoknak már létezni kell. Vond be felülettel a szükséges helyeket, az eredményt ellenőrizd le a teljes képernyőre nagyított perspektíva ablakban, **Shaded view** módba.

Ha véletlenül olyan helyen hoztál létre felületet, ahol nem szándékoztál, akkor térj át Pick Faces módba és a csúcspontjain keresztül jelöld ki az ominózus felületelemet majd a Delete-tel töröld le azokat. Így a határoló élek és a pontok megmaradnak.

Kinyomás térben

Jóni jó, de van-e jobb? Hisz ennek a vonalzonak nincs vastagsága. Nosza, adjunk neki. Ez sem egy bonyolult dolog, van rá külön funkció. Mielőtt alkalmaznánk, gondoljuk át, mit kell tennünk. A tárgyunknak van már kiterjedése X és Z irányban, csak az Y vastagság hiányzik. Ha ezen tengely mentin kinyomnánk, meglenne ezen kiterjedése is. Pontosan ezt csinálja az ajánlott funkció.

Válaszd ki a vonalzót, majd használd az **Object menü Mold** pontját. Egy kérdező jelenik meg hét kapcsolóval, amelyek mindegyike egy-egy funkciót jelképez. Ezen funkciók közül most az **Extrude**-ra

Detail editor

van szükség, klikkelj rá a nevére vagy a kapcsolójára. Újabb kérdező jelenik meg, teli input mezőkkel és kapcsolókkal. Az Extrude kinyomást jelent a funkció a kiválasztott tárgyat nyomja ki **annak Y tengelye** mentén. A kinyomott tárgyra a kinyomás mentén "bőrt" feszít, azaz oldalait összefüggő felülettel vonja be, mintegy csövet, vagy rudat képezve belőle. Az extrudálás történhet egyenes mentén, vagy egy görbe vonalat, másnéven utat követve.

Előbbit a "**To Length**", utóbbit az "**Along Path**" kiikszelésével érhetjük el. Mivel utat még nem tudunk létrehozni, maradjunk az első esetnél, amúgy sincs most másra szükségünk.

A kinyomódás hosszát a **Length** input mezőben lehet megadni. Ide írd olyan értéket, ami arányos a vonalzó hosszához, szélességéhez. Ez lesz a vastagsága.

A **Path** input mezőbe kellene írni az út nevét, ha az **Along Path**-t kapcsoltuk volna be.

A **Sections** a kinyomás hosszában létrejövő szekciók száma, maradjon most ez 1.

Ha csavarmenetet, csavart oszlopot, vagy efféléket akarnánk készíteni, erre is volna mód. A program a kinyomás művelete alatt képes elforgatni a tárgyat az **Y tengelye** körül. Az elforgatás mértékét kell fokokban beírni az **Y Rotation** input mezőbe.

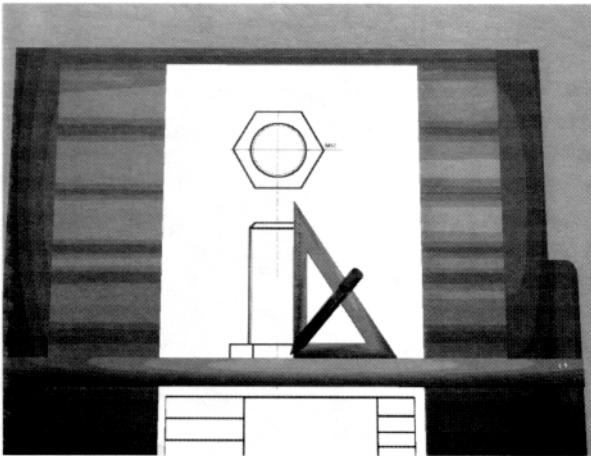
Az **X Scaling** és **Z Scaling** segítségével a tárgy a kinyomás művelete alatt megváltoztathatja a méretét. Egy korong ilyenén történő kinyomásával csonka kúpot hozhatnánk létre. Ha a pisai Ferde Toronyot akarnád megmintázni, ezt is lehet. Az **X** és **Z Translate** input mezők segítségével miközben kinyomod a tárgyat, oldal irányban el is tolhatod. Az eltolás mértékét kell ide beírni Imagine egységben. Majd később próbáld ki egy Disk primitívvel!

Kinyomás térben

A **Mirror Ends** kapcsolóról még nem esett szó. Ha ezt kiikszeled, a kinyomott tárgy utolsó szekciója tükörképe lesz az első szekciónak. Erre nemsokára mutatok egy példát.

Az itt nem ismertetett kapcsolói az **Extrude Data** kérdezőnek az út mentén történő kinyomásra vonatkoznak, máskor ejtünk róla szót.

Nyomjuk tehát ki a vonalzókat 10 egység mélyen, egy szekcióból. Egy valódi vonalzó hosszabbik átfogója le szokott ferdítve lenni, még ezt csináljuk meg rajta. Menj át felülnézetbe és Pick Points módba. A kérdéses él hosszában két pontot beljebb kell mozgatni néhány egységgel. Felülnézetben ez a két pont egybeesik, Drag Box-al, shiftelve könnyedén kiválasztható mindkettő. Igen ám, de a megfelelő helyen nincs rácspont, amire rá lehet igazítani azokat. Szerencsére a rácssűrűséget tudjuk szabályozni. A Grid Size kérdezőjében adj meg 10 egységet. (Annyit, amennyit a kinyomás mélysége volt.) Így már nincs akadálya, hogy leélezzük a vonalzókat. Az előlnézetben azonban újabb probléma merül fel, a vonalzó csúcsánál. Ennek kiküszöbő-



Detail editor

lését rád bízom kedves olvasó, de annyit tanácsolok, hogy vagy a Lock-ot kapcsolj ki, vagy vedd még sűrűbbre a rácsot.

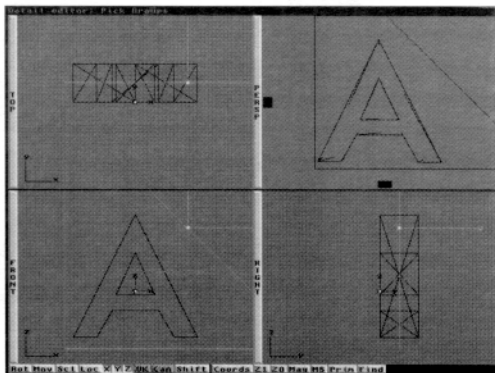
Kivágások

Következő gyakorlatként készíts el egy nagy "A" betű alakzatot a front nézetben. Adjunk ennek a betűnek felületet. Belátható, hogy az előző módszerrel lassú és nehézkes lenne ilyen és bonyolultabb tárgyak felülettel való ellátása. Kellene valami gyorsabb módszer, amellyel automatizálni lehetne legalább egy részét a feladatnak. Szerencsére van ilyen, a program ki tud vágni két tárgyat egymásból. Ha az egyik tárgy a betűnek körvonalala, a másik pedig egy sík, már kész is a felülettel ellátott betű.

Az egész eljárást úgy kell elképzelni mint a tézstaszaggatást. A tézstaszaggatónak magasságának is kell lenni, tehát a művelet előtt nyomjuk ki térben a körvonalát. A kinyomási hosszának nincs nagy jelentősége, légyeg hogy vastagabb legyen mint a sík, amint kivágunk vele.

Hozzuk létre a síkot is . A vízszintes és függőleges szekciók száma 1 legyen. A Scale és a Move segítségével állítsd úgy be a síkot, hogy a betű keresztülmegyen az egyik felületelemén. Ezzel beállítottuk a tézstaszaggatónkat, már csak utasítani kell a programot, hajtsa végre, amit kigondoltunk.

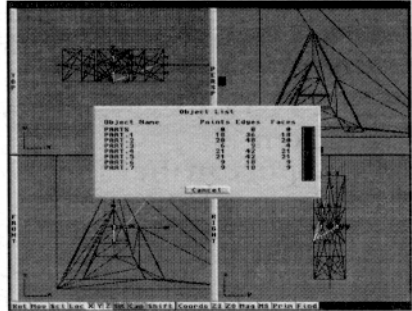
Válaszd ki mindkét tárgyat, majd használd az **Object**



menü **Slice** pontját. A gép rövid időre gondolkodóba esik. Amíg el nem készül a szeleteléssel, a **Cancel** gombra kattintva visszakoehatsz.

Miután kész a szeletelés, több új tárgy is létrejön, nem csak a betű vágja szét a síkot, hanem a sík is a betűt. Nézd meg a Find Requester-ben, hány tárgy van a szerkesztőben. Nekünk csak a kivágott betűalakra lesz szükségünk.

Ha megnézed a tárgyakat, észreveheted, hogy az egyes részek tengelypontjait világos vonalak kötik össze. Ez egy eddig még nem ismertetett struktúra jele, teszünk egy kitérőt, hogy pótoljuk ezt a hiányt.



Csoportok

A tárgyak amelyekkel eddig találkoztunk, mind önálló, másoktól független tárgyak voltak, egyetlen tengellyel. Amikor két tárgyat a Join vagy Merge segítségével összekapcsoltunk, teljesen eggyé váltak, közös tengelyük lett. Az összekapcsolt tárgyakat később csak nehézkesen tudnánk újból alkotóikra szétszedni, ráadásul az elemek egyéni tulajdonságaikat elvesztik. Ezt kiküszöbölhetjük, ha a tárgyakat, amelyeket más tárgyahoz akarunk kapcsolni, nem fűzzük fizikailag is egybe, hanem logikai csoportot, stuktúrát alkotunk belőlük. A csoport bizonyos szempontból egy tárgyként viselkedik, egyben mozgatható, kimenthető, stb, de ha akarjuk, bármikor hozzáférhetünk az egyes elemeihez, amelyek önálló tárgyak.

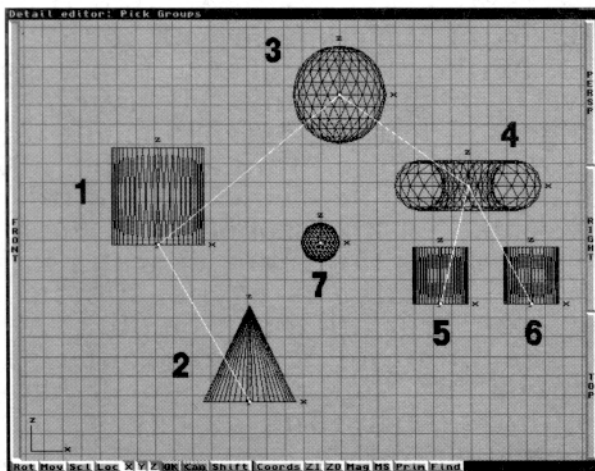
Detail editor

A csoport egy meghatározható hierarchia szerint épül fel, minden tárgynak megvan a saját helye a csoporton belül. Az egész szerkezetet úgy képzeld el mint egy fát, vagy a tartalomjegyzék stuktúráját a lemezeken.

A csoporton belül minden tag kétféle státuszú lehet: **szülő** vagy **gyermek**. Némelyik tárgy egyszerre viseli mindkét státuszt.

A szülő státusz azt jelenti, hogy a tárgyhoz alárendelt tárgyak, gyermekek kapcsolódnak. Egy szülőnek több gyermeke is lehet. Minden szülő egyben gyermek is, hiszen valakitől származnia kell. Ez alól csak egy kivétel van, az ősszülő, aki a csoport hierarchiájának legtetején áll.

A gyermek státusz azt jelenti, hogy a tárgy egy hierarchikusan magasabban lévő tárgyhoz kapcsolódik, mintegy abból származik. Minden gyermek félárva, azaz csak egy szülője, közvetlen felmenője lehet, mint ahogy egy fájl is csak egy tartalomjegyzékben lehet egyszerre. Másik szülő alá legfeljebb csak másolata tartozhat, de az már nem ugyanaz.



Minden gyermek lehet egyúttal szülő is, belőle is származhatnak gyermekek.

Nézzünk meg a következő képen egy stuktúrát:

Itt az 1-es henger az ősszülő, belőle származik a 2-es kúp és a 3-as gömb. Ez a gömb egyben szülő is, gyermeke a 4-es gyűrű, amelynek két gyermeke van, az 5-ös és 6-os hengerek. A 7-es gömb nem tartozik a csoporthoz.

Tegyük most félre egy kicsit a vonalzónkat. Menj Pick Group módba és a Find Requesterben válaszd ki a **Parts** menü tárgyat, amely 0 pontból, 0 élből és 0 felületből áll. Ez egy tengely, amely a szeleteléskor létrejött csoport ősszülője, ezen keresztül tudjuk a teljes csoportot egyben kezelni.

Láthatod, az egész narancs színűre vált, most használd a Pick Selectet, amivel kiválaszhatod őket. Mentsd ki az egészet majd töröld le a szerkesztőből. Csoportokat kezelni a **Pick Groupban** lehet!

Hozd létre a front nézetben azokat a tárgyakat, amelyek az előző képen szerepeltek és vidd is őket azokba a pozíciókba. Ezeket gyakoroljuk egy kicsit a csoportkészítést és kezelést. Tárgyak csoportba fűzése egyszerű dolog, mindössze ki kell választani a csoportosítandó tárgyakat, majd használni az **Object menü Group** pontját. Bonyolultabb a helyzet, ha a hierarchiát is szabályozni akarjuk.

Válaszd ki az 1-es hengert, majd utána shiftel kúpot és a 3-as gömböt. Most használd a Group-ot. A csoport létrejön, az ősszülő, egyben az egyetlen szülő a henger lesz. Mindíg az lesz a szülő, amelyik tárgyat elsőnek választottuk ki.

Tegyük most a 3-ast szülővé, rendeljük alá a gyűrűt! Válaszd ki a gömböt, mire a vele csoportban lévő másik két tárgy kiválasztottsága megszűnik. Multival válaszd mellé a gyűrűt is, majd csoportosítsd őket.

Detail editor

Alakul. Most az előző módszerrel rendeld a gyűrű alá a két kisebb hengert. Ezzel kész a struktúra, amit a képen is látsz. A java még csak ezután jön!

Válaszd ki ismét az 1-es hengert. Mivel ez az őszülő, az egész csoport kiválasztódik. Másodiknak válaszd a 7-es számú kis gömböt és csoportosítsd össze őket. A gömb a henger alá rendelődik.

Az **Undo**-val vond vissza a csoportosítást. Most a kúpot majd a kis gömböt választva csoportosítsd össze őket. A gömb a kúp alá rendelődik. Ismét **Undo**. Idáig nem volt semmi különös, bármelyik szülőhöz bármikor rendelhetünk újabb tárgyakat.

Próbáljuk meg fordított sorrendben, válaszd ki előbb a kis gömböt, majd a nagy hengert és csoportosítsd őket. Az eredmény látszólag ugyan az, valójában azonban nem. Most a 7-es lett az őszülő, az 1-es ennek a gyermeke. Próbáld ki, klikkelj a kis gömb tengelyére, az egész csoport kiválasztódik, míg a henger tengelyével csak a tőle lefelé eső tárgyakat tudjuk befolyásolni. Megint Undo.

Próbáld most ki a gömb-kúp kombinációt csoportosítani. Az eredmény egy figyelmeztetés, miszerint a tárgy már csoportban áll. Ezt az okozta, hogy a kúp felé egy második szülőt akartunk létrehozni, ami tudvalevőleg nem lehetséges. Nincs mit tenni, mint tudomásul venni ezt az OK-val.

Mi van, ha mi mégis a gömböt szeretnénk a kúp fölé és csak afölé a hengert? Szét kell bontani a kúp-henger csoportkapcsolatot. Szétbontáskor ki kell jelölni a lebontandó csoport szülőjét, majd használni az **Object menü Ungroup** parancsát.

Jaj! A teljes csoport struktúra szétbomlott! Mivel nem ezt akartuk gyorsan Undo-zzuk el. Mi történt az előbb? A teljes csoportot kijelöltük őszülőjén keresztül, az Ungroup emmiatt a teljes csoportra hatott.

Akkor most mi lesz, újra fell kell építeni a csoportot? Szerencsére nem, ezért is használtuk az Undo-t. Válaszd ki a **Mode** menüből a **Pick Objects**-et. Ebben a módban tudunk a csoport egyes elemeihez mint önálló tárgyakhoz hozzáférni. Ha előtte ki volt választva a teljes csoport, akkor most csak az összülője a henger marad kiválasztva. Próbáld újra az Ungroup-ot. Most csak azok a csoportkapcsolatok szűntek meg, amelyek ebből a tárgyból indultak. Így már nem olyan nehéz felépíteni a kívánt struktúrát.

Rakd össze újból a csoportot úgy mint a kiinduláskor volt. Térj vissza Pick Group-ba és válaszd ki az összülőt, a hengert. Most próbáld ki a Move interaktív transzformációt. A csoport minden tagja saját befoglaló keretével helyettesítődik, az egész csoport együtt mozgatható.

Tedd le a csoportot és válaszd most ki a nagy gömböt. Nem csak ez, hanem a belőle származó teljes struktúra is kiválasztódik. Ha most megpróbálsz őket mozgatni, ezek együtt mozdulnak, míg a csoport többi, feljebbi vagy más ágon lévő tagjai helyben maradnak.

Próbáld meg ezt úgy is, hogy a gyűrűt, vagy annak valamelyik gyermekét vállaszkod ki. Megállapíthatjuk, hogy a teljes csoport csak az összülön keresztül választható ki. A struktúrában alacsonyabban lévő szülőt kiválasztva a belőle származó struktúra is kiválasztódik, míg a felmenői vagy a vele párhuzamos ágon lévők nem. Ilyen esetben a szülő kvázi összülőként viselkedik. Arra is van lehetőség, hogy több külön ágat egyszerre kijelölj és manipulálj.

Fontos, hogy az előbb ismertetett kijelölő műveletek nem változtatják meg a csoport szerkezetét. Persze, ha a kiválasztott alcsoportot vagy tárgyat töröljük, az más, akkor megváltozik a csoportszerkezet.

Mi a helyzet a Pick Objects-el? Nagyjából hasonló,

Detail editor

csak itt a tárgyak önállóan választhatók ki és manipulálhatók. Ha a struktúra közepéről törölünk egy szülőt, a gyermekei nem esnek ki a csoportból, hanem a törölt tárgy szülője alá rendelődnek, megtartva a lefelé menő struktúrájukat.

Ennyit a csoportokról (egyenlőre). Az hogy ilyen sok helyet szenteltem ismertetésüknek, nem véletlen. Megértésük igen fontos, a Cycle editort például nem is tudnánk enélkül használni.

Kivágás, másolás

Töröld le a tárgyakat és tölts vissza az A betűt. Bontsd szét a csoportot elemeire, majd keresd meg kivágószerszámunk által létrehozott síkszeletek közül azt, amelyik a betűt formázza. Legkönnyebben ezt a **Select Next**-el tudod megtenni. Találsz a tárgyak között félbevágott kivágószerszámot, sítot amelyből kivágták egy A betű körvonalát, stb. Ha megtaláltad amelyiket kerestük, válaszd ki. Valahogy ki kell emelni a többi közül, ezt megtehetjük mozgással is, de inkább mutatok egy másik módszert.

Az object menüből válaszd ki a **Copy** vagy a **Cut** parancsok valamelyikét. Ezek egy átmeneti tárolóba másolják a tárgyat, amely bármekkora lehet (amíg a memória engedi). Az átmeneti tároló egyszerre csak egy bemásolást tartalmazhat, a következő ilyen művelet törli az előzőt. Az egy bemásolás nem föltétlen jelent egy tárgyat, ha egyszerre többet, jelölsz ki, mindegyik bekerül az átmeneti tárolóba, de nem lehet külön kezelni őket.

A különbség a két parancs között annyi, hogy a Cut, miután végrehajtotta a bemásolást, törli a tárgyat a szerkesztőből, míg a Copy változatlanul hagyja azt. A Cut, Copy csak tárgyakon vagy csoporton

Pontok mozgatása, másképpen

tokon működik, a tárgyakat alkotó pontokat, éleket és felületeket ily módon nem lehet önállóan manipulálni.

Ha bemásoltad az A-t, a **Pick All** funkcióval válassz ki minden tárgyat és a **Delete**-tel töröld le azokat. Ezután az **Object menü Paste** pontjával másold vissza az elraktározott tárgyat. Ez az eredeti pozíciójában fog megjelenni.

A betűnk, mivel síklapból vágtuk ki, sík, de ezen könnyen változtathatunk a már ismert **Extud**-dal. A **Slice** funkció nem csak síklapból való kivágásra használható, hanem térbeli alakzatok közötti szeletelésre is. Elvileg. Gyakorlatilag nagyon rigolyás. Ha a kivágásban résztvevő alakzatoknak valamely éle egybeesik, nem tudja vérehajtani a műveletet. Ilyenkor néha segít, ha valamelyik tárgyat kissé arrébbtoljuk.

Pontok mozgatása, másképpen

Folytassuk a tárgykészítési eljárásokkal való ismerkedésünket, készítsünk egy négyzet alakú gúlát. A kiindulási alap egy 2×2 szekcióból álló Plane primitív. Van valakinek ötlete erre? Igen, jó megoldás, hogy az előlnézetben kiválasztjuk a középső pontot, majd a felülnézetben a move segítségével kialakítjuk a gúlát. Azonban én mást javaslok. Lépj be a Drag Points módba. Ez hasonló a Pick Points-hoz, de itt a kiválasztott pontokat rögtön el is tudjuk mozgatni, "fogd és vidd" elv szerint.

Mivel az előlnézetben nem tudjuk a kérdéses pontot mélységében mozgatni, vagy a felül, vagy a jobb nézetben kell ezt megtenni. Nosza!

De mi történt? Hisz nem a középső pont mozdult el, hanem a szélső, én meg blamálok itt magam az

Detail editor

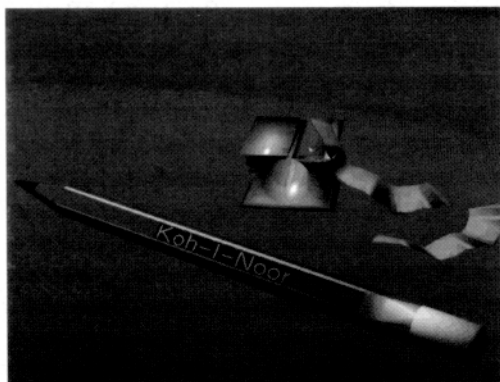
új módszeremmel. Ha belegondolunk, ezen nézetekből a középső pontot nem is lehet látni, mivel a síklap élén lévő pontokkal egybeesik, így azok takarják. Akkor most mit tegyünk? Maradjunk a bevált mozzgatásnál. Ez egy konzervatív vélemény, és mint ilyen, a haladás gátja. (Ez meg demagógia, de ne politizáljunk!)

Hallom, van aki azt javasolja, forgassunk egy kicsit a síkon az Y tengelye körül, így a belső pont kikerül a takarásból. Ez már jobb, de így sem tanulunk semmi újat.

Ha kifogytatok, akkor elmondom az én ötletemet. A shiftet nyomva tartva, az előlnézetben válaszd ki a síklap középső pontját. Most menj át a felül, vagy oldalnézetbe és miközben nyomva tartod a shiftet nyomd le az egér bal gombját, de ne legyen a pointer a tárgy egyik pontja fölött sem! Most engeddd el a shiftet, és máris mozgathatod az előzőleg kijelölt pontot vagy akár pontokat. Amíg a shift nyomva van, a Drag Point-tal nem lehet pontokat mozgatni, csak kiválasztani. Tudom, ennél mégis csak egyszerűbb lett volna a move, de akkor nem tanultuk volna meg ezt a nagyszerű funkciót. Képzeld csak el egy több tucat, vagy akár több száz pontból álló tárgyat,

amelyet a pontok áthelyezésével akarunk megformálni. Brrr! Belegondolni is rossz abba a sok move-ba!

Néhány oldallal korábban ígertem egy példát az **Ext-rude** funkció **Mirror Ends** kapcsolójára, most ez következik.



Készíts egy kúpot, alap paraméterekkel. Ezt fogjuk tükrözve kinyomni. Készítünk belőle egy mindkét végén hegyes rudat, mondjuk egy kétvégről hegyezett ceruzát.

Mielőtt nekikezdenénk, a kúp tengelyét el kell forgatni, mert az Y nem a hossz tengely, mi pedig azon irányba akarjuk a ceruzát nyújtani. Gondolom a Transformation-nal nem okoz gondot ez a művelet.

Most jön az **Extrude**. Nyomjuk ki a kúpot 500 egység hosszan, a **Mirror Ends** bekapcsolásával. Az eredmény önmagáért beszél.

Feliratok

Kicsit korábban készítettünk egy A betűt. Az egész ABC elkészítése ezen a módon kissé hosszadalmas és fáradságos művelet lenne, szerencsére ez is automatizálható. Az Imagine rendelkezik egy beépített funkcióval, mellyel bitmap fontokat tudunk átalakítani Imagine object formátumba. Ennek menete a következő:

Válaszd ki a **Functions menü Addd** pontjának **Font object** alpontját. A program rövid ideig el lesz magával, ezalatt listát készít a rendszer **Fonts:** könyvtárában lévő fontokról és azok méretéről. Miután ezzel megvan, egy kérdező jelenik meg, benne a meglévő fontok és méreteik listája, gördítőléccel.

Az alattuk lévő Text input mezőben adhatjuk meg a konvertálandó szöveget. Válassz ki egy jó nagy betűtípust (minél nagyobb, annál szebb lesz az object) és írd be a szöveget. A kérdező alján van egy eddig még nem látott nyomógomb, a **Read**. Ez újraolvassa a Fonts:-ot. Akkor hasznos, ha közben Shell-ből megváltoztattuk a Fonts:-hoz tartozó

Detail editor

tartalomjegyzéket az **Assign**-nal. Persze csak Amigán van erre lehetőség, PC-n nem működik a multitasztk.

Ha mindez megvan, nyomd meg az OK-t. Egy kérdező jelenik meg, a következő kérdéssel: Adsz felületet a tárgynak? Ha No-val válaszolsz, csak a körvonal jön létre, ha Yes-szel, felülete is lesz a feliratnak. Az így kapott tárgynak nincs mélységi kiterjedése, de ezen gondolom tudsz változtatni.

Ha nem mondtam volna, az egész felirat egyetlen tárgy lesz.

Gyorsított megjelenítés

Egy hosszabb felirat, vagy sok tárgy jelenléte a szerkesztőben nagyon lelassítja a képfrissítést. Általában nincs is szükség arra, hogy minden tárgyat teljes részletességében lássunk, elég ha látjuk hol helyezkedik el. Erre is van funkció, a **Functions** menüben lévő **Quickdraw**. Meghatározhatjuk, hogy mely tárgyakat ne rajzolja ki a program, teljes részletességgel, hanem helyettük az interaktív műveletekből már ismert befoglaló keretekkel ábrázolja azokat.

A **Quickdraw All** alkalmazásával minden tárgy az említett dobozzal helyettesítődik, a **Quickdraw Pick** csak az éppen kiválasztott tárgyakra hatásos. A **Quickdraw None** segítségével térhetünk vissza a teljes megjelenítéshez. A **Quickdraw** a perspektíva nézetre nem vonatkozik, ott mindig minden tárgy teljesen kirajzolódik.



Utak és alkalmazásuk

Az **Extrude** funkció ismertetésekor kihagytuk az út mentén történő kinyomást és a hozzá kapcsolódó opciókat. Most ezek ismertetése következik.

Gondokodtál-e azon, hogyan készült a program főcímlogo-ja? Természetesen ezt is Imagine-nel készítették, nem is bonyolult módon. A logo egy font object út mentén történt kinyomásával jött létre. Az út oldalnézetből egy nyújtott S-betűt formázott.

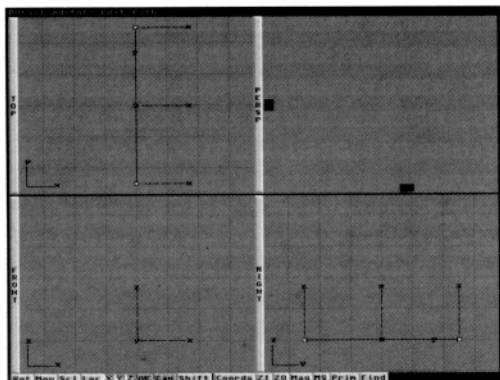
Csináljunk mi is egy ilyen tárgyat. Először készítsük el az utat. Ehhez azonban legalább az út szerkezetének alapjaival tisztában kell lennünk. Az út egy görbevonala, teljesen más felépítésű, mint azok a tárgyak, amelyekel idáig találkoztunk. Az útnak nincs felülete, a kész képen soha nem látható, mindig csak folyamatok vezérlését végzi. Egyetlen közös az útban és az egyéb tárgyakban, hogy ennek is van egy saját tengelye.

Mint említettem az út egy görbe, nincs pontja, éle, hanem vezérpontjai, más néven kontroll pontjai vannak, amelyeken a görbe meghatározott irányban halad keresztül. Egy görbének minimum két vezérpontjának kell lenni.

Kétféle úttípust különböztetünk meg, nyitottat és zártat. A nyitottnak két végpontja van, míg a zártnak egy sem, az önmagába tér vissza. Az útnak iránya is van, az első ponttól az utolsóig mutat. Az, hogy a két végpont közül melyik az első, fontos, mivel a tárgy amelyik követi az utat, az első ponttól indul és az utolsó felé tart.

Ennyi előtörténet után lássuk hogyan hozhatunk létre utat a Detail editorban. Két megoldást is kínál az Imagine, az egyik a Functions menü Add pontjának **Open Path** és **Closed Path** pontjai. Ezek

Detail editor



nyitott és zárt utat hoznak létre. Kísérletezzünk most az elsővel.

Az Open Path egy Y irányba mutató szakaszt hoz létre, melynek tengelyrendszere a szakasz közepére esik. Ahhoz, hogy az utat alakítani tudjuk, át kell térnünk **Edit Path** módba. Ezt, mint minden módot a **Mode** menüből választhatjuk ki. Ebben a módban az utunk két végén egy-egy tengely jelenik meg, tengelyponttal. Ezeket a tengelyeket nem szabad összekeverni a tárgy tengelyével, ezek a kontroll pontok tengelyei. A görbe mindig a kontroll pontok tengelyének Y irányán halad keresztül.

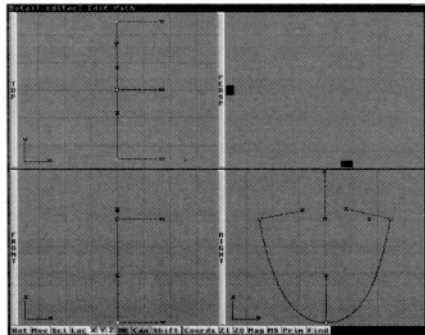
Hogyan tudjuk alakítani az utakat? Azok kontroll pontjait kell manipulálni. Mindhárom transzformáció típust, a mozgatót, méretezést és forgatót tudjuk használni, mind interaktívan, mind a Transformationon keresztül. Próbáljunk alakítani ezen az úton. A felülnézetben válaszd ki az alacsonyabban lévő pontot. Frontnézetben ez az, amelyik közelebb van hozzánk. Nyomd le az r billentyűt és kapcsolj be a forgatót az Y tengelyre. Forgasd el a kontroll pont tengelyét kedved szerint. Látszólag semmi sem történt, valójában az út hosszában elcsavarodott. Ha most végig követtetnénk az utat egy tárggyal, bekapcsolhatnánk azt az opciót, hogy a tárgy kövesse az elcsavarodást is. Ennek most nincs jelentősége, csak mutattam mik a lehetőségek.

Próbálj az útból egy U alakú görbét hajlítani. A már kiválasztott pontot állítsd vissza eredeti irányba

Utak és alkalmazásuk

(például Undo-val) , majd az oldalnézetben fordasd el az X tengely körül az óramutató járása szerint kb 90 fokkal. A másik pontot is fordasd el ugyan ezen tengely körül, de ellenkező irányban, Túl széles az U nyílása, ezért a move segítségével vidd egymáshoz közelebb a pontokat. Amit kaptunk, az még nem igazán U, inkább egy tál alakja. Kellene egy újabb pont, amivel meghatározhatjuk a magasságot, illetve a mélységet.

Válaszd ki ismét az első pontot. Mellesleg ez az út First Point-ja, tehát innen indul az út. Most használd a **Functions** menüből a **Fracture**-t, minek hatására az út közepén egy új kontroll pont jön létre. Ha a másik pontot választottad volna, a Fracture-nak nem lett volna hatása, mivel ez a pont az utolsó, belőle nem indul görbe, csak érkezik bele. A Fracture mindig azt a szakaszt felezve hoz léte újabb vezérpontot, amelynek az első pontját választottuk ki. Ha most a görbe első pontját kiválasztva használod a Fracture -t , a görbe első negyedében jelenkezik új pont, míg ha a középső, új pontot, akkor a harmadik negyedben. Egyszerre több görbeszakasz is megfelelhető.



Az újonnan létrehozott kontroll pont tengelye úgy helyezkedik el, hogy az út alakjában nem történik változás. Ha ezt az új pontot a move transzformációval lehúzd, a görbe egész jól megközelíti az elképzelt formát.

Az esetlegesen fölöslegessé vált kontroll pontokat, kiválasztásuk után a **Delete**-tel le lehet törölni. Az út

Detail editor

utolsó két pontját ellenben nem engedi törölni a program.

A zárt utak kezelése semmiben sem tér el a nyitottakétól, ezért azokat most nem részletezzük.

Említettem, hogy van más módja is az utak létrehozásának. Töröld le az előző kísérletünk maradékát, majd hozz létre három tengelyt a **Functions/ Add/ Axis**-szal. Rakd szét őket a front nézetben, egy képzeletbeli háromszög csúcsaiba. Válaszd ki mindet, tetszőleges sorrendben. Most aktiváld az **Objects menü Make Path** pontját. Ezzel a művelettel létrehoztunk egy utat. Az új tárgy tengelye az elsőnek kiválasztott Axis lett. A tengelyekből lettek a kontroll pontok. Az út első kontroll pontja az elsőnek kiválasztott Axis helyén, annak az irányát felvéve keletkezett.

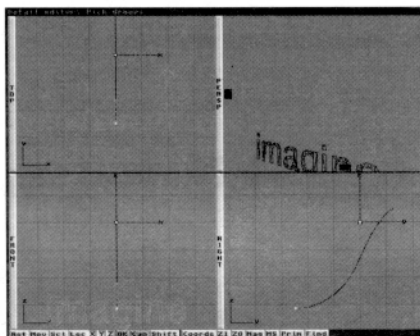
Fontos! A tárgy tengelye és a First Point szétváltak, mostmár teljesen függetlenek egymástól. Az út tengelyét elmozgatva az első kontroll pont változatlanul marad!

Próbáld különböző sorrendben összekötni a tengelyeket. Megállapíthatsz ennek eredményéből egy fontos dolgot: Az út a kontroll pontokon mindig az Y tengelyen megy keresztül a negatívtól a pozitív irány felé tartva. Ez fontos dolog, segítségével például könnyedén meghatározhatjuk melyik a First Point és milyen az út iránya.

A **Make Closed Path** a kiválasztott tengelyekből zárt utat hoz létre. Itt is az elsőnek kiválasztott Axis lesz a tárgy tengelye és ezzel azonos a First Point is, de a FP egyúttal az utolsó pont is, ide tér vissza a görbe. A zárt út irányát is meg tudjuk határozni előző tapasztalatunk segítségével.

No profi útászok, lássunk neki a logokészítésnek! Mivel már mindent tudunk az útról, amit tudni kell, nem okoz gondot a képen látható görbe elkészítése.

Az **Add Font Object**-tel hozd létre a feliratot, a **Scale** segítségével pedig arányosítsd egymáshoz az út és a felirat méretét. Az út mentén történő kinyomáskor lényegtelen a tárgy és az út egymáshoz viszonyított pozíciója, mindig a tárgy tengelypontja halad végig az úton, követve annak elfordulásait. Miután a tárgy tengelye (és vele együtt a tárgy is) végighaladt az úton, a tengely (csak a tengely) visszatér a kiindulási pozícióba. Erdemes tehát a tárgyat az út közelébe vinni, hogy ne essen tőle messze saját tengelye a kinyomás után.



Nézd meg az út nevét a **Find Requester**-ben. Válaszd ki a tárgyat és lépj be az **Extrude** kérdésőjébe. Kapcsold be az **Along Path**-t. A **Path** input mezejébe írd be az út nevét.

Az **Along Path** bekapcsolásával két új kapcsoló aktiválódott. Ezek csak az út mentén történő kinyomáskor működnek.

Ha azt akarjuk, hogy a tárgy, miközben végighaladt az úton, kövesse is a tengelyeinek orientációjával annak fordulásait, be kell kapcsolni az **Align Y to Path** kapcsolót. Ha ez nincs bekapcsolva, egy kanyarodó úton haladó tárgy nem fordul el, hanem a kanyarban oldalazni, csúszni kezd. Az utat, hozzá igazodva követő tárgy Y tengelye mindig úgy áll, hogy érintője legyen az utat alkotó görbének, X és Z tengelyei pedig 0 fokban zárnak be az út azonos tengelyeivel.

A **Keep X horizontal**-nak csak az előző opció bekapcsolásakor van jelentősége. Hatására az X

Detail editor

tengelye a tárgynak megtartja eredeti orientációját, miközben az Y tengely továbbra is az út érintőjeként viselkedik. Gyakorlatilag ez annyit jelent, hogy az út hossztengelye körüli csavarodást nem követi a tárgy.

Nekünk most csak az első opcióra van szükségünk. Amíg egyenes mentén nyomtuk ki a tárgyakat, elég volt egyetlen szekció is, de most íven történik ez a művelet, legalább 5 részből kell állni a kinyomott tárgynak. Bonyolultabb út esetén még több szekcióra lenne szükség, hogy szebbé tegyük az ívet, de ez drasztikusan emelné a fájl méretét is.

A **Mold**-on belül van egy művelet amelyik nagyon hasonlít az Extrude-ra. A neve **Replicate**. Ez is térben, a tárgy Y tengelyének irányában, megadott távolságra, vagy út mentén nyomja ki a tárgyat, de a szekciók nem kötődnek össze, a tárgy nem lesz egybefüggő. Az eredeti tárgyról másolatok jönnek létre és a távolság, vagy az út mentén egyenletesen helyezkednek el. Az eredmény, függetlenül a szekciók számától mindig egy tárgy.

A Replicate kérdezője majdnem azonos az Extrude-é-val. Egyetlen különbség van, hogy a Mirror Ends nem aktív.

Ez a funkció kitűnően használható ismétlődő elemek létrehozásához, mint például a vasúti sín talpfái.

Tárgyak szétbontása

Egyetlen tárgy több példányban történő szétmásolásához ismerünk már két funkciót is. Most próbáljuk megoldani a következő feladatot. Egy tárgyat, mondjuk egy gömböt kell elkészíteni öt példányban, egymástól 100 egységre. Fontos megkötés, hogy mind az öt gömb külön tárgy legyen.

Mágneses hatás

Utóbbi feltétel miatt a Replicate szóba sem jöhet. Marad a Copy és a Paste. Ezzel viszont körülményes a feladat megoldása.

Minden várakozással ellentétben most nem egy új tárgytöbbszöröző funkciót mutatok be, hanem egy olyat, amellyel tárgyakat darabolhatunk szét.

Készíts egy gömb primitívet 25 egység sugárral. Válaszd ki ezt, majd lépj be a Replicate kérdésőbe. Többszöröld meg a gömböt, 500 egység hosszú egyenes mentén 4 szekcióból. A tárgy nem felel meg annak a feltételnek, hogy önálló tárgyra van szükség, de ez csak múltó állapot. Menj át **Pick Points** módba és válaszd ki az egyik gömb összes pontját, de más pontot ne! Most válaszd ki a **Functions** menüből a **Split**-et. A tárgy szétválik, a kiválasztott pontok új tárgyat alkotnak.

A Split-tel tehát szétszethetjük a tárgyat elemeire. Ha olyan pontot választottunk le, amelyiket él, esetleg felület is kötött össze egy maradandó ponttal, az él és a felület megszűnik. A létrejövő tárgyak tengelye mindig egybeesik, az eredeti pozícióban.

Ha precízek akarunk lenni, nem a kiválasztott pontok válnak le a tárgyról, hanem a ki nem választottak. Ez abból látható, hogy a Find Requester-ben a kiválasztott pontokból keletkező tárgy tartja meg az eredeti nevet.

Mágneses hatás

A következő tárgy, amit elkészítünk, egy kapuív lesz. Hozz létre egy tengelyt és az **Add Lines**-szal próbálj meg kialakítani egy ívet. Nem akarok senkit megsérteni, de ahogy elnézem a legtöbb ív siralmasra sikeredett. Való igaz, kézzel szabályos ívet rajzolni

Detail editor

igencsak nehéz. Bemutatok most egy módszert, amivel ez a probléma megoldható.

Töröld le az előző próbálkozás eredményét, állíts be a rácssűrűséget 10 egységre, kapcsold be a **Lock**-ot és az előlnézetet váltsd teljes képernyősre. Most hozz létre egy új tengelyt és az **Add Lines**-szal készíts egy 220 egység hosszú vízszintes vonalat, úgy, hogy minden rácspontba teszel egy pontot. Ez összesen 23 pontot jelent. Ezekből a vonalakból alakítunk ki a **Drag Points**-szal egy szabályos ívet.

Ha a már ismert módon használod a funkciót, semmivel sem lesz könnyebb ívet készíteni mint az előbb. Válaszd most ki a **Functions menü Magnetism** pontjának **On/Off alpont**ját, ami egy kapcsoló menüpont. Ha ki van pipálva, akkor a **Drag Points**-ban megragadott pontokon kívül néhány mellettük lévő pont is mozdul, mintha a megragadottak mágnesként vonzanák őket.

A mágnesesség paraméterei szabályozhatók. Ehhez a **Magnetism menü Setup alpont**ját kell kiválasztani. A megjelenő kérdésőben a következő paraméterekeket adhatjuk meg.

Radius of Influence. A mágneses hatás sugara. A megragadott pont mint középpont körül ekkora sugarú gömbön belüli pontokra hat a mágnes.

Minimum Radius (Random Mode) - Ha a kérdező ajánl lévő **Random Radius**-t bekapcsoljuk, a mágnesesség sugara minden művelemben véletlenszerűen alakul, a **Minimum Radius**, és a **Radius of Influence** értéke között. Mindkét esetben **Imagine** egységben kell a sugarat megadni.

Percent at Radius (0...100). A mágnesesség erőssége a távolság függvényében. Az mutatja meg, hogy a megfogott ponttól legmesszebb lévő, a hatásos gömb szélén álló pontra a központi mágneses erősségének hány százaléka jut. Ha 0, az

Mágneses hatás

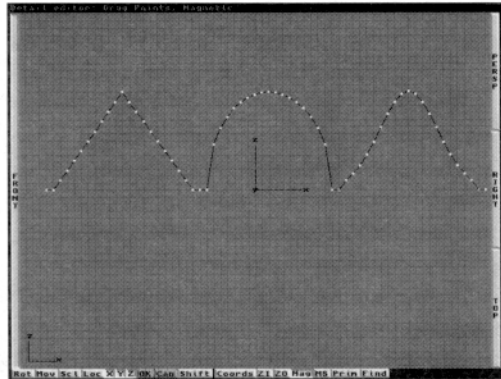
utolsó pont már nem is mozdul. Ez adja a legfinomabb ívet. Ha értéke 100, az utolsó, pont is akkora mértékben mozdul, mint a megfogott. Ebben az esetben még a mágneses hatás alakja (lásd később) sem érvényesül.

Az, hogy a mágnesesség milyen alakban terjedjen, beállíthatjuk a **Magnetism Type** három kapcsolójával.

A **Cone** bekapcsolásával a mágnes vonzásában lévő pontok kúpot formáznak.

A **Done** egy kupolát készít a mágneses hatás alatt álló pontokból.

A **Bell** segítségével harang alakban hat a mágnes.



A kérdező utolsó kapcsolóját már említettem. Ezt kiikszelve amikor megfogunk egy mozgatandó pontot, akkor a mágneses hatás sugara felvesz egy véletlen értéket a **Radius of Influence** és a **Minimum Radius** határai között. Ezt főleg felszíni alakzatok, hegyek készítésénél használjuk.

Ezen ismeretek birtokában egy kis gyakorlás után könnyedén megoldhatjuk az ívkészítést és a hasonló problémákat.

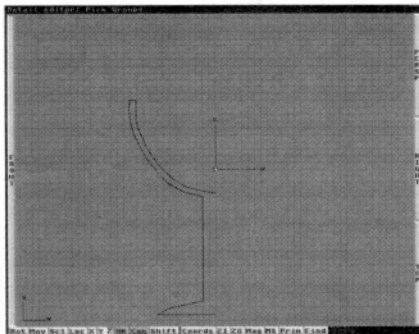
Egy tanács a Magnetism-hez. Amikor használod a Lock legyen kikapcsolva, mert a rácstraigazítás tönkreteszi az ívet.

Detail editor

Forgástestek készítése

A most következő részben megtanulunk forgástesteket készíteni. Az első munkánk legyen egy pezsgős-pohár. A forgástestet úgy készíthetjük el, hogy megrajzoljuk a körvonalát, majd a test középpontja körül megforgatjuk azt. Ha 360 fokot forgatunk a körvonalon, elég csak az egyik oldalát elkészíteni.

A tervezett pohár körvonala tehát a következő képen néz ki a front nézetben:



A poharunk kehely részét a magnetism segítségével alakíthatjuk ki. A kehely kettős vonala a falvastagság miatt szükséges. Egyéb helyeken ez úgysem látszik, ezért is nem készítettem el. A középső pontot a tengely vonalára kell helyezni, hogy ne legyen lyuk a poháron. Ezt elérhetjük úgy is, hogy az iv kialakítása után visszakapcsoljuk a Lock-ot, de van rá másik mód is.

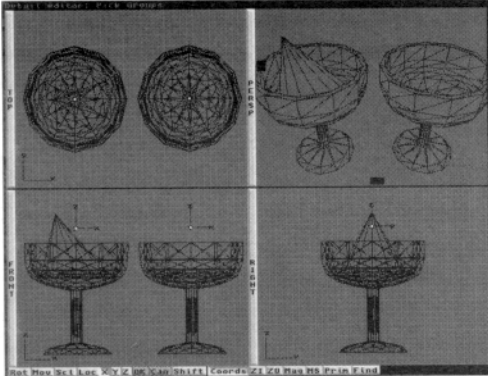
Vidd a megfelelő rácspont közelébe ezt a pontot, majd **Pick Point**-ban válaszd ki. Ezután használd a **Functions** menü **Snap to Grid** pontját, aminek hatására a kiválasztott pont vagy pontok mindegyike a neki legközelebbi rácskereszteződésre ugrik.

Ha már itt tartunk, bemutatok egy másik "pontugraltató" műveletet. Sokszor lehet arra szükség, hogy a tárgy néhány pontját egy egyenesbe rendezzük. Erre van egy külön funkció. Válassz ki egyszerre legalább három pontot a tárgyon majd használd a **Functions**-ból a **Taut**-ot, a kiválasztott pontok egy

Pontok kitakarása

egyenesere rendeződnek. Az egyenest az elsőnek és az utolsóknak kiválasztott pont határozza meg. A pontok között lévő távolság arányosan változik.

Ha kiválasztod ezt a kupa körvonalán, Undo-zd el a változást és válts át **Pick Group** vagy **Pick Object** módba. A kiválasztott körvonalat most megforgatjuk a **Mold** kérdezőben lévő **Spin** vagy **Sweep** funkciók valamelyikével. Ezek nagyon hasonlóak egymáshoz,



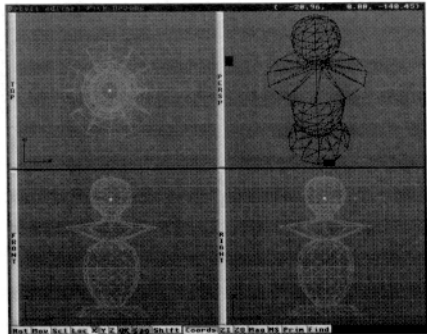
paraméterkérdőjük is azonos. Mindkettő a Z tengely körül forgatja meg a tárgyat a **Spin Angle** input mezőben szögfokban megadott mértékben, annyi szekcióból, amennyit a **# of Sections**-ban megadtunk. A megforgatott körvonalat rögtön

felülettel látja el. A két funkció között egyetlen különbség van, míg a **Sweep** esetében a forgatás minden pontra vonatkozik, addig a **Spin** a körvonal első és utolsó pontját nem forgatja, de felülettel köti a tengelyhez, zárt tárgyat hozva létre.

Pontok kitakarása

Gyakorlás képpen készítsd el a következő sakkfigurát.

A gyalog gallérját szándékosan készítetem ilyen nagyra, ennek kisebbitése lesz a feladat. Próbáljuk ezt kb. akkorára venni,

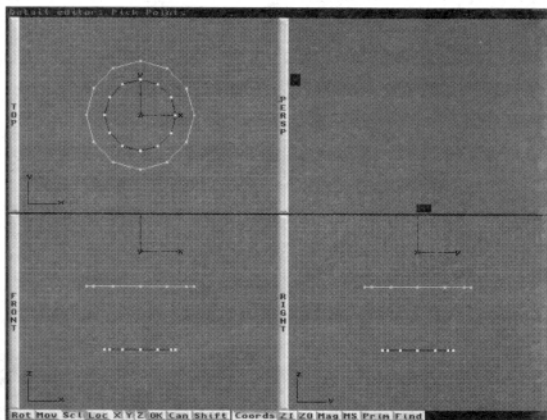


Detail editor

mint a teste. A megoldás egyszerű, az előlnézetben Pick Points-ban Drag Box-szal kiválasztjuk a gallér élét alkotó pontokat, majd a Scale segítségével megváltoztatjuk az általuk meghatározott kör átmérőjét. A segédrács segítségével könnyen beállíthatjuk a kívánt méretet. Ha ezt a műveletet nem csak körülbelülre, hanem pontosabban akarjuk végrehajtani, ez már nem biztos hogy elég. Ha bezoomolunk a nézetbe, nem látjuk egyszerre a gallért és a testet. Legjobb volna, ha a felülnézetben tudnánk egymásra helyezni a pontokat, ez azonban nem lehetséges, mert ott nagy a zsúfoltság.

Az Imagige ilyen és hasonló esetekre rendelkezik egy funkcióval. A szerkesztésben részt nem vevő pontjait a tárgynak kitakarhatjuk, így azok nem zavarhatnak. Ezt a **Mode** menü **Hide Points** módjának bekapcsolása

után tehetjük meg. Az ebben a módban kiválasztott pontok látszólag eltűnnek. Miután kitakarjuk a felesleges pontokat a hozzájuk tartozó éllekkel együtt, folytathatjuk a munkát a tárgyon.



Az elrejtett pontok csak **Pick Group** vagy **Pick Object** módba térve rajzolódnak ki ismét.

Fogaskerék és lobogó

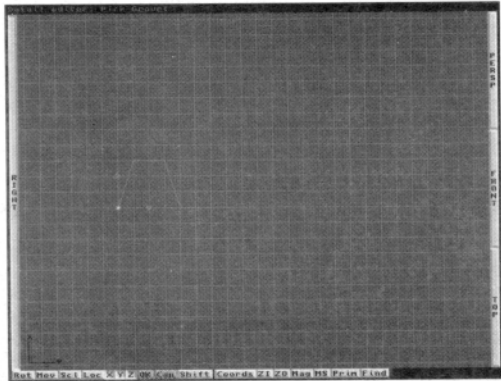
Fogaskerék és lobogó

Gondolnátok, hogy a fogaskereket és a zászlót hasonló módon lehet elkészíteni? Nézzük hogyan.

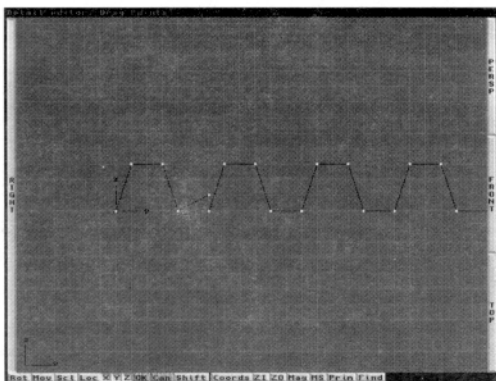
A kereket nem korongból fogjuk kialakítani, mert a fogak egyenkénti kialakítása lassú és pontatlan. Helyette mindössze egy fogat készítünk el, azt a **Replicate**-tel többszörözzük egyenes mentén, majd a fogaslécet egy képzeletbeli henger köré tekerve létrehozzuk a fogaskerék körvonalát. Ezt ellátjuk felülettel és az **Extud**-dal kialakítjuk a vastagságát.

Az oldalnézetben készíts el egy fogat a fogaskerékből. Műszaki szakemberek elkészíthetik a fogvolvenst is, mi maradjunk az egyszerű trapéznál. A képen látható méretarányokat mindenképpen tartsd be, hogy jó legyen a kerék alakja.

Egy későbbi műveletben a tengely helyzete is fontos lesz, már most úgy építsd fel a tárgyat, mint a képen látszik. Azért kellett oldalnézetben elkészíteni a fogat, mert a **Replicate** a tárgy Y tengelye mentén fogja a többszörözést végrehajtani. Álljon a kerék 30 fogból, a **Sections**-ba tehát 29-et kell írni (a 30. az eredeti lesz). A kinyomás hossza 29* a fog alapjának szélessége. Nálam egy fog 120 egység széles, a kinyomást így 3480 egység hosszán kell végrehajtani. Mivel a későbbiek során még lesz mit számolgatni,



Detail editor



javaslom te is ebben a méretben készítsd el a kereket. A képen 20 egység a rács-méret.

A **Replicate** végrehajtása után a fogak látszólag egybefüggőek, valójában azonban nem. Ahol a két szomszédos fogelem összeér, ott nem egy, hanem két különálló pont van. Erről meggyőződhetsz, ha Drag Point-tal elhúzod az egyiket. A fogaskerekünk szempontjából jobb volna, ha a fogak láncolata folytonos volna, össze kell tehát fűzni ezeket a pontokat mind a 29 helyen.

Nem bonyolult a művelet, **Drag Box**-szal, shiftelve válaszd ki sorban a pont párokat és kapcsold össze azokat a **Functions** menü **Join**-jával. Ha a Join-nal pontokat kapcsolunk egybe, a közös pont mindig az elsőnek kiválasztottal lesz azonos. Ennek akkor van jelentősége, ha nem fedésben lévő pontokon használjuk.

Az összefűzést nagyon meggyorsítja, ha a Join-t a jobb Amiga J billentyűzet kombinációval rövidítjük.



Most jön a beígért számolgatás. A 30 fog együttesen $30 \cdot 120$, azaz 3600 egység hosszú. Ez a képzeletbeli henger kerülete, a sugara ebből adódóan $3600/2\pi$, ami 572.9578 egységre jön ki. A hengerre feszítést a tárgy Z tengelyével párhuz-

Fogaskerék és lobogó

zamos, attól "r", (sugár) távolságra lévő tengely körül hajtja végre a program. A tárgy Y tengelye sugárirányba mutat.

Ezen ismeretek birtokában forgasd el a tárgy tengelyét, hogy a hengerre feszítést végre tudjuk hajtani.

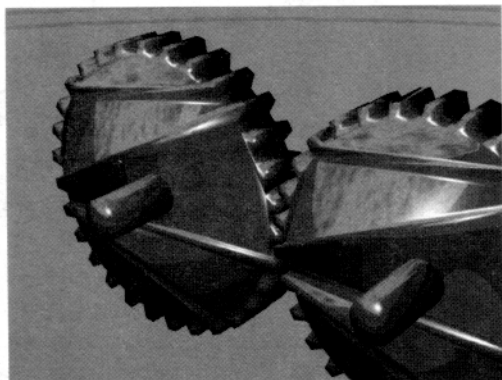
Most hívd elő a Mold kérdezőjét és válaszd a **Conform to Cylinder** funkciót. A megjelenő kérdésben mindössze két input mező van.

A **Cylinder Radius** a képzeletbeli henger - amire a tárgyat feszítjük - sugara. Ezt éppen az előbb számoltuk ki, írd be az 572.9578-at.

A másik mező, az **Object Width**, a hengerre csavarandó tárgy szélessége. A mi esetünkben ez a fogasléc hossza. Mivel itt szélességértéket vár a program, a fogasléc hosszának csak a felét kell beírni. Ezt a szélességet mindkét irányba alkalmazva fogja kapni a valódi hosszt.

Bizonyos esetekben ide nem a valós értéket szoktuk írni, hanem egy arányosítottat. Például ha csak félkörívet akarunk készíteni, kétszer akkorának adjuk meg a szélességet, ellenben ha két fordulatot akarunk a hengerre tekerni, fele szélességet kell beírni. A mi esetünkben egy teljes fordulatra van szükség, de nem többre, ezért a szélességnek 1800-at adjunk meg. Ezután csak annyi van hátra, hogy a **Perform**-ra kattintva feltekerjük a fogasívünket.

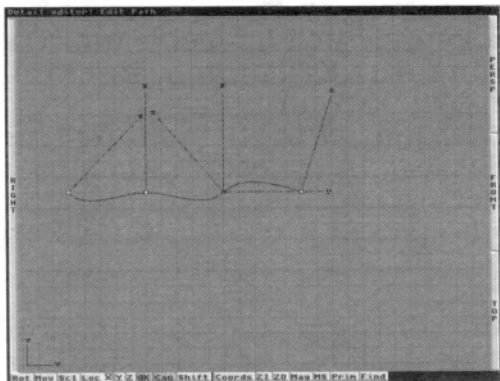
Csak egy klikkelés, ugye milyen egyszerű? Egy kicsit



Detail editor

azért bonyolultabb. A fogasív két végén lévő pontokat, amelyek fedésbe kerültek, a **Join**-nal fűzd össze és készíts a tárgynak felületet, majd mélységet. Az egésznek a betetözése a **Quick Render** lesz, de ezt is rádbízom.

Biztos feltűnt, hogy a **Mold**-ban három **Conform...** funkció van. A **Conform to Sphere** szinte azonos az előbbivel, kérdezője is majdnem olyan. A különbség,



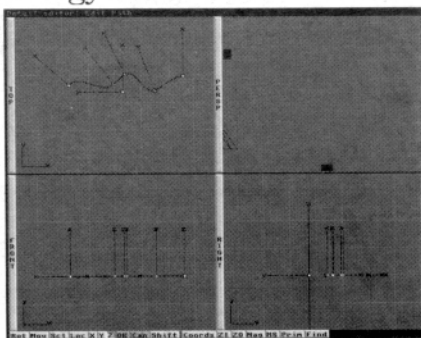
hogy itt nem egy képzeltbeli hengerre, hanem gömbre feszítjük fel a tárgyat, tehát nem csak a Z, hanem a X tengelye körül is megtekerjük. A **Sphere Radius** a gömb sugara, az **Object Radius** pedig a tárgyé. Mindazok, amit a **Conform to**

Cylinder-nél leírtam, itt is érvényesek, ezért ezzel a funkcióval most nem foglalkozunk részletesen.

A harmadik ráfeszítő funkció a **Conform to Path**. Ezzel a kiválasztott tárgyat egy útra feszíthetjük. Próbáljuk ki, készítsünk vele egy zászlót.

Hozzunk létre egy több kontroll pontból álló utat, ami valahogy így néz ki.

Hozz létre egy Plane primitívet, a mérete nem számít, de vízszintesen jó sok, mondjuk 50 szegmensből álljon, függőlegesen elég egyből. Rakd a görbe mellé, mint itt.



Repülőgépszárny, hajótest és társaik

A kiválasztott síkot a **Mold Conform to Path** funkciójával ráfeszítjük az útra. A kérdezőben mindössze egyetlen input mező van, itt az út nevét adhatjuk meg.

Ha elsőre (vagy később másodjára) nem jönne be az amit elképzeltél ne csüggedj, ez az Imagine legtitokzatosabb funkciója. Itt egyáltalán nem számít a tárgy tengelyének pozíciója és iránya, csak a tárgy és az út egymáshoz viszonyított helyzete. Általában a legjobb, ha az út és a tárgy tengelyeinek orientációja azonos. Kísérletezz vele, hogy kitapasztald.

Repülőgépszárny, hajótest és társaik

A számítógépes modellezésben az egyik legnépszerűbb tárgy a repülőgép. A gép szárnyát és testét nem könnyű elkészíteni, csakúgy mint egy csónak modelljét. Szerencsére a Detail editor egy speciális funkciója segítségünkre van.

Amikor a csónakkészítő csónakot épít, először a bordázatot faragja ki és állítja össze. Erre a bordázatra rakja fel utána a palánkot.

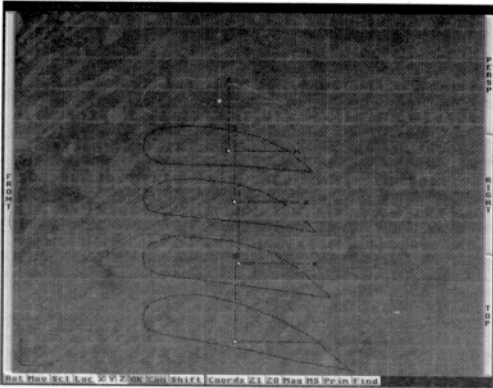
A repülőgép törzsét és szárnyát is hasonló eljárással készítik. Az előbb említett funkció szintén ezt teszi. Az elkészített bordázatra felülettel, "bőrt" feszít.

A bordázatra és az egész eljárásra van néhány megkötés. A bordák általában csak körvonalak, bonyolult tárgyakat erre a célra nem szoktunk alkalmazni.

A bordáknak mind azonos számú pontból kell állni és a pontok sorrendje is fontos. Az egyes bordák azonos sorszámú pontjait éllel köti össze a funkció, majd az így létrejött tárgyat felülettel vonja be.

Lássunk erre egy gyakorlati példát. Hogy megfeleljünk annak a feltételnek, hogy az egyes bordák

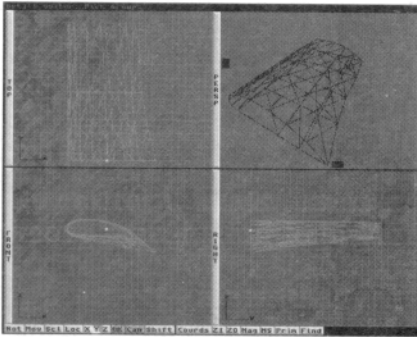
Detail editor



azonos pontokból álló körvonalak, egy tárgyból induljunk ki, azt másoljuk szét a szükséges számokban. Én korong primitíveket szoktam használni, amelyeknek a középső pontját törölöm.

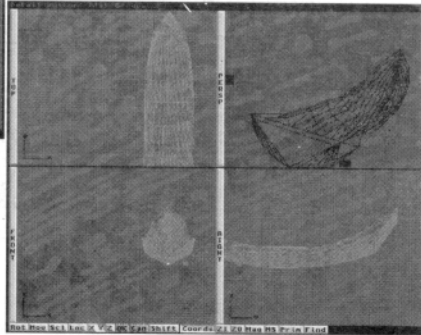
Alakítsd ki a frontnézetben az alábbi öt szárnybordát.

Természetesen, amikor a bordákra felfeszítjük a vásznat, azokat nem így rendezzük el, hanem egy



vonalban, mélységben eltolva, de akkor egymást zavarnák a tárgyak a kialakításkor. Ha kész minden borda, mozgasd egy magasságba azokat és

rendezd el a szárny hosszában. Válaszd ki őket sorban. A sorrend fontos, mivel a vásznat ilyen sorrendben fekteti fel a funkció. Az összevissza kiválasztott bordákat kissé furcsán burkolná be.



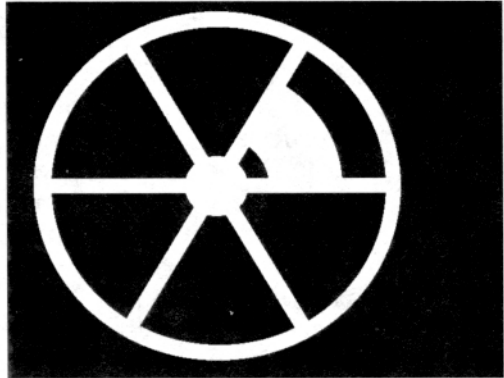
A kiválasztott tárgyakra a burkolaltot, a "bőrt" az **Object** menü **Skin** pontjával feszíthetjük fel.

Rajzból tárgy

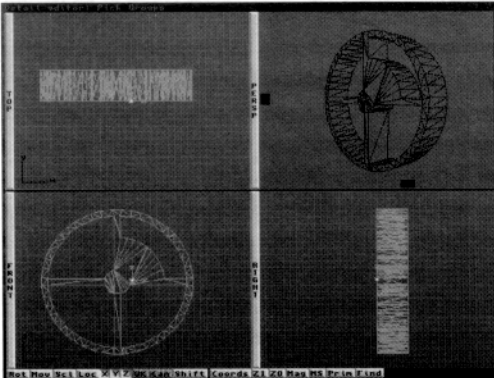
Hasonló módon készíthejük el a gép törzsét is. A csónak kialakításakor mindössze annyi a különbség, hogy a borda nem zárt, de a funkció csak az összefüggő részekre feszít bőrt, ezért technikailag ez nem okoz gondot.

Rajzból tárgy

A következő feladat, tervezzük meg egy mozdony kerekét, küllőkkel, ellensúllyal. A már ismert módszerekkel ezt az alakzatot elkészíthetjük, én azonban mutatok egy új, kényelmesebb és gyorsabb módszert.



Valamilyen bitmap-es rajzolóprogrammal készítsd el a fenti képet két színnel, 640*512 (vagy 640*480) pixel méretben. Ha kész, válaszd ki az **Object** menüből a **Convert IFF/ ILBM** menüpontot. A meg-



jelenő fájlselektorban válaszd ki a képfájlt. Rövid töltés után egy kérdező jelenik meg, ahol választhatunk, adjunk-e a létrejövő tárgynak felületet. A felület a rajzszínnel készített ábrán jön létre, a háttéren nem. Ha a felület

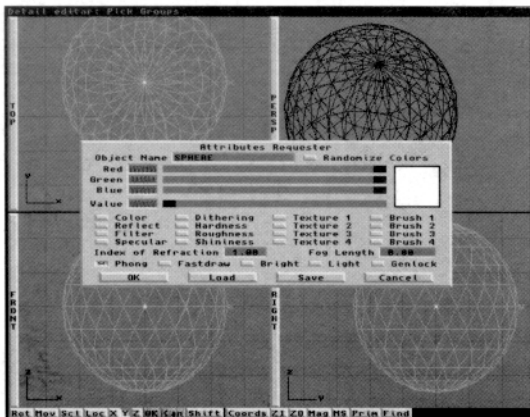
Detail editor

készítést választjuk, a gép kis gondolkodás után elkészíti a tárgyat, amelyet tetszés szerint tovább alakíthatunk.

Attribútumok

Aki idáig követte a könyvben leírtakat, elmondhatja magáról, hogy majdnem mindent el tud készíteni az Imagine-ben. Egy nagyon fontos dologról azonban még nem volt szó: A tárgyak felületi tulajdonságáról, úgymint szín, tükrözőképesség, felületi érdesség, stb. Gondolom ennek hiányát mindenki érezte már, hisz a **Quick Render**-rel elkészített képeken minden tárgy fehér színű volt. Ez nem a QR hibája, hanem a tárgyak alaptulajdonsága. Amíg meg nem változtatjuk, minden újonnan létrehozott tárgy fehér színű.

Nosza akkor színezzünk. Ezirányú kísérleteink lefolytatásához készítsünk egy gömb primitívet. Válaszd ki ezt, majd az **Object** menüből aktiváld az **Attributes** funkciót. Egy jókora kérdező fog elének tárulni, rengeteg beállítási lehetőséggel. Ismerkedjünk meg mindnyájukkal.



A legfelső input mező az **Object Name** tartalmazza a tárgy nevét. Ezzel a névvel azonosítjuk a tárgyat a **Find Requester**-ben. Az input mezőben természetesen megváltoztathatjuk ezt. A név teljesen füg-

getlen a tárgyat tartalmazó fájl nevétől, így PC-n is írhatunk be ékezetes betűt a max. 17 karakteres névbe.

A **Randomize Colors** kapcsolóval véletlen színeket adhatunk a tárgyaknak. Ennek a menete a következő: A kapcsolóra kattintva egy **Random Seed** nevű kérdező jelenik meg, egyetlen input mezővel. Ebben egy tetszőleges, max. 80 jegyű számot adhatunk meg, amelyet a véletlenszám-generátor fog felhasználni. Az Enter lenyomása után ez a véletlen generátor a tárgy színéből kiindulva a tárgy minden egyes felületelemének más színt ad. Absztrakt képek elkészítésénél nagyon jól használható.

Az **Attributes Requester** innen több részre osztható fel. Közvetlenül az előbb ismertetett mező és kapcsoló alatt négy input mező és tolóka található **Red, Green, Blue** és **Value** feliratokkal. Ezekkel lehet az egyes értékeket beállítani. Hogy melyikeket, azt a lentebb elhelyezkedő kapcsolókkal választhatjuk ki. A tolokák mellett, jobbra egy négyzetben a beállított színértékeket követhetjük nyomon.

A tolokák alatti kapcsolók négy csoportra vannak bontva, csoportonként négy-négy kapcsolóval. Ezek közül egyszerre többet is használhatunk, sőt, akár mindet.

Az első oszlopban a tárgy színével kapcsolatos dolgokat találjuk. Amikor innen valamelyiket kiválasztjuk, a **Red, Green** és **Blue** tolokákkal és input mezőkkel adhatjuk meg a hozzájuk tartozó értékeket, színösszetevőnként külön-külön.

A második oszlopban a felületi minőség beállítási lehetőségeinek kapcsolói vannak. Ezek mindegyike a **Value**-n keresztül kaphat értéket.

A harmadik oszlopban az Imagine egyik nagy erősségének a matematikai textúráknak kapcsolóit találjuk.

Detail editor

A negyedik, utolsó oszlop kapcsolóival tudunk képeket a tárgyra feszíteni és ezekkel másosítani bizonyos tulajdonságaikat.

A kérdező alsó harmadában még találunk két input mezőt és öt kapcsolót, majd legalul négy, ismerős nevű gombot.

A rövid áttekintés után sorra vesszük a tárgyak egyes attribútumait, azok jelentőségeit, lehetőségeit.

A **Color** gondolom nem szorul bővebb magyarázatra. A kapcsolóra klikkelve a **Red**, **Green** és **Blue** input mezőkben adhatjuk meg a tárgy színét. Az eredményt láthatjuk a jobb oldali négyzetben. A tárgy színe 24 biten definiálható, de a szín ablakban csak egy géptől és megjelenítőtől függő színelőzetest kapunk.

Bizonyos esetekben a tárgyak színe nem az lesz a kész képen, amit itt beállítottunk. Egy tárgy színe úgy alakul ki, hogy a ráeső fénysugár bizonyos színösszetevőit elnyeli, míg más színeit visszaveri. A visszavert színekből áll össze a tárgy látszólagos színe. Ha a fény nem, vagy csak részben tartalmaz olyan színű fénysugarakat, amelyet a tárgy visszaverni képes, hamis színeket kapunk.

Azt hogy a szín milyen színű sugarakat és milyen mértékben tükrözzön, külön is megadhatjuk a **Reflect**-tel. Ezt szintén színösszetevőnként külön-külön lehet definiálni. Mivel a tükröződő tárgyakban környezetük látszik, a magas Reflect érték felülbíráhatja a tárgy beállított színét!

Fontos! Valódi tükröződést csak **Trace** algoritmussal számol a program! **Scanline** képszámításnál a tükröződő felületekben mindössze a **Background Picture**, **Sky Brush Global Sky** és a **Ground primitív** látszik. Egyébb tárgyak helyett csak fényfoltokat tesz a program a tükröződő tárgyra.

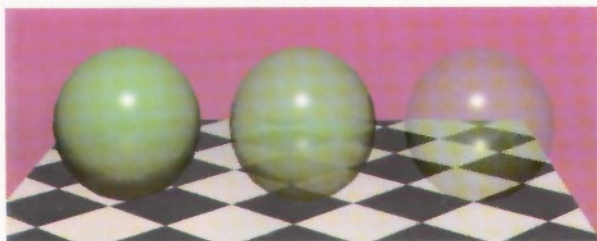
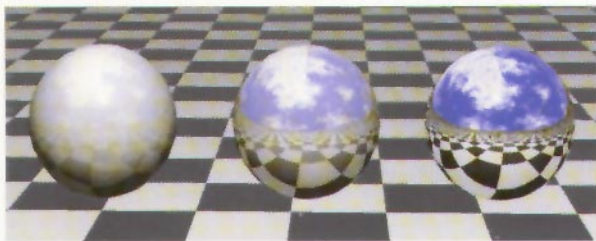
Egy tárgy nem csak elnyelheti vagy visszaverheti a

Attribútumok



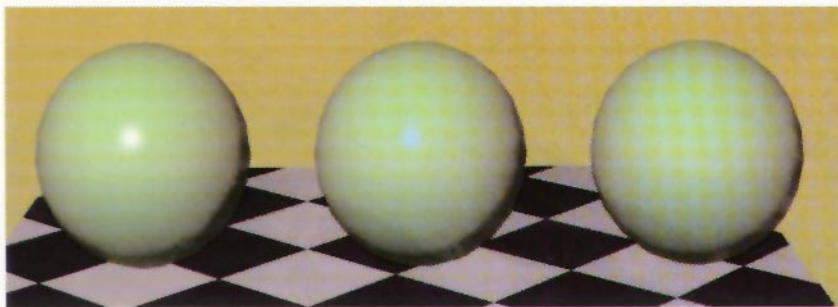
A Randomize Color bekapcsolása után minden felületelem egyedi, véletlen színt kap. Gyakorlati haszna nem sok van, de absztrak képeknél jól jöhet ez a funkció. A kialakuló színeket nem tudjuk közvetlenül befolyásolni.

A Reflect attribútum a tárgy fényvisszaverőképességét szabályozza. A képen az első gömb Reflect-je 0, a másodiké 100, míg a harmadiké 200, a többi érték az alapon maradt. Valódi tükröződés csak ray-traced módban számolódik.

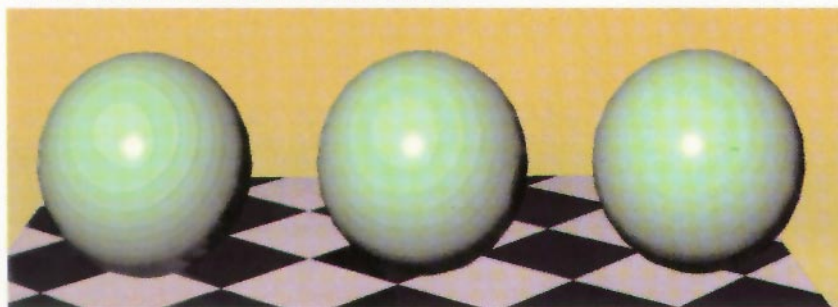


A Filter paraméterrel a tárgyak átlátszóságát állíthatjuk be. A képen az egyes Filter értékek sorban: 0, 100, 200. A tárgyak többi attribútuma, a szín kivételével, az alapértelmezés szerinti. A teljesen átlátszó tárgy nem látható!

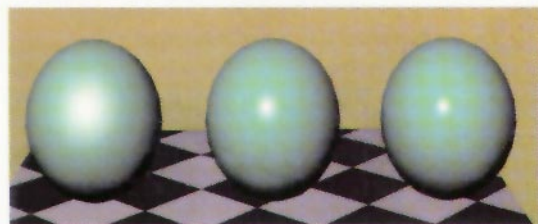
Detail editor



Specular. Ez az attribútum a tárgyon megcsillanó fény színéért felelős. Értéke az első gömbön 255, 255, 255, a másodikon 0, 0, 255, míg a harmadikon 0, 0, 0, azaz nincs csillanás.



A Dithering paraméternek a nem 24 bites képek készítésekor van jelentősége. Értéke minél nagyobb, a program annál jobban elsimítja a színátmeneteket. A képen a Ditheringek balról-jobbra 0, 120, 255.



A Hardness paraméterrel a tárgyak keménységét lehet állítani. A képen lévő tárgyak 0, 120 és 255 keménységűek.

fényt, hanem át is engedheti. Ennek mértékét a **Filter**-rel állíthatjuk be.

Mivel a tárgy színét a visszavert színek határozzák meg, maximális filter értéknél, vagyis mikor minden fényt átenged, a tárgy nem látható.

Az idáig ismertetett attribútumok, az Object Name kivételével, nem csak tárgyakra, hanem felületelemekre önállóan is megadhatók. Ennek menete igen egyszerű. **Pick Faces** módban ki kell jelölni a kívánt felületelemeket, majd meghíni az **Attributes**-t. Például a gömbre csíkot rakhatunk.

Ez így önmagában még nehézkes eljárás, hisz nem egyszerű ismételt kiválasztani ugyanazokat a felületelemeket. Elsőre eltalálni egy színbeállítást szintén ritka. Szerencsére a program készítői gondoltak erre és létrehoztak egy funkciót, amellyel a tárgy tetszőleges felületelemeit alcsoportokba, **subgroup**-okba köthetjük. Ez nem azonos a **Group**-pal, azok tárgyak között jönnek létre. Amikor a kiválasztott felületelemeket alcsoportba kötjük, egy alcsoportnevet kell megadni. A művelet a tárgy szerkezetére nincs hatással, de a későbbiek során az alcsoportnévre hivatkozva könnyedén tudjuk újból kiválasztani ugyanazokat a felületelemeket.

Lépj most ki az **Attributes Requester**-ből, hogy kipróbálhassuk a subgroupképzést és a hozzá kapcsolódó funkciókat. Válassz ki a próba gömbön néhány tetszőleges felületelemet **Pick Faces** módban. Ezután használd a **Functions** menü **Make** pontjának **Make Subgroup** alpontját. Egy **Subgroup name** input mezőt tartalmazó kérdező jelenik meg, ahol meg kell adni az alcsoport nevét. Ha kész, zárd le enterrel. Egyszerre tetszőleges számú alcsoportot hozhatunk létre a tárgyon, sőt az egyes felületek több alcsoportba is tartozhatnak. Alcsoportot létrehozni csak **Pick Faces** módban lehet. Ha ké-

Detail editor

sőbb ki akarjuk választani valamelyik alcsoportot, a **Pick/Select** menüből aktiválni kell a **Pick Subgroup**-ot. Erre egy olyan kérdezőt kapunk, mint a Find Requester, de itt nem a tárgyak, hanem az alcsoportok nevei vannak és a tárgy pontjainak valamint éleinek száma nincs feltüntetve, csak a felületei.

Alcsoportot megszüntetni a **Functions/ Make/ Unmake Subgroup**-pal lehet. Ez nem törli a csoportba tartozó felületeket, csak az alcsoportba tartozásukat ejti.

Ha azt akarjuk, hogy az alcsoport felületei ne legyenek kiválasztva, például egy tárgy minden felületelemét, kivéve az alcsoportot akarjuk kiválasztani, a **Pick/Select** menü **UnPick Subgroup** pontját kell alkalmazni. Ez csak az alcsoport tárgyainak kiválasztását szünteti meg. Sajnos ez a funkció nem minden változatban működik rendesen, a kiválasztást megszünteti ugyan, de a színek módosításával ezt nem jelzi, a felületek továbbra is a kiválasztottságot jelölő színűek maradnak.

A **Pick Subgroup** és az **Unpick Subgroup** nem csak **Pick Faces**, hanem **Pick Points** és **Pick Edges** módban is használható. Ekkor egy kérdezőben választhatunk, hogy az alcsoport felületelemeinek határoló vagy belső éleit vagy pontjait, esetleg mindkettőt akarjuk-e kiválasztani. Az **Interior Edges** és **Points** a belső éleket és pontokat, míg a **Boundary Edges** és **Point** a határoló éleket és pontokat jelenti, az **All** pedig mindet. Más funkció is segíti a felületek, de akár az élek vagy pontok kiválasztását is. A **Pick Select** menü **Pick Range** pontját **Pick Points**, **Edges** vagy **Faces** módban használhatjuk. Ekkor egy input mező jelenik meg, ahol megadhatjuk, hogy hányadik ponttól /éltől/ felülettől kezdve hányadikig akarjuk kiválasztani és

milyen lépésközzel.

Ennek megértéséhez tudnunk kell azt, hogy az *Imagine object* formátumban sorban tárolja a pontokat, éleket és felületeket. A kiválasztáskor ezt a sorrendet veszi figyelembe.

Próbáld ki ezt a funkciót, válaszd ki egy alap paraméterekkel létrehozott gömb minden harmadik felületét és töröld azokat, majd *Quick Render*-rel készítsd el a képet.

Ezekben a módokban is használhatók a **Pick/Select** menü **Pick All**, **Home**, **Select Next**, **Select Prev** és **Sort** pontjai. A pontok, élek és felületek rendezését a **Sort**-tal el tudjuk végezni, az *object* sorrendnél megismert módon, vagyis a szükséges sorrendben kiválasztjuk a rendezendő elemeket, majd használjuk a **Sort**-ot. Ez nem változtatja meg a tárgy fizikai szerkezetét.

Az **Unpick Range** a **Pick Range** fordítottja, a sorozatba eső elemek kiválasztását szünteti meg. Ennyi.

Az utolsó attribútum, ami a színekkel kapcsolatos, a **Specular**. Ez a testen megcsillanó fény színe és intenzitása. Minél nagyobb ez az érték, a tárgy annál csillogóbbnak tűnik. Ez a csillogás nem jelent tükröződést is, csak fényfoltokat láthatunk a csillogó helyeken.

A második attribútumcsoportba a felületi paraméterek tartoznak, értéküket a **Value**-val állíthatjuk be.

A **Dithering** egy színkeverési eljárás. Akkor van rá szükség, ha nem 24 bites képeket generálunk. Ebben az esetben rendelkezésre álló színekészlet (a 4096, vagy 256 szín) nem elegendő a tárgy elkészítéséhez, színvonalak jönnek létre. A program össze tudja mosni ezeket a vonalakat, szebb színátmenetet

Detail editor

eredményezve. Minél nagyobb a Dithering értéke, annál nagyobb fokú a színösszemosás.

Kemény, fémes kinézet eléréséhez alacsony, míg lágy felületekhez magas érték használatát javaslom. 24 bites képek készítésénél a Ditheringnek nincs jelentősége, mivel ekkor elegendő szín áll rendelkezésre a színátmenetek elkészítéséhez.

A tárgy keménységét a **Hardness**-szel állíthatjuk be. Minél keményebb egy tárgy, annál nagyobb a rajta csillanó fényfolt méret. A 0 érték jelenti a legpuhább tárgyat. Ebben az esetben hiába állítottuk be a **Specular**-t, azt figyelmen kívül hagyja a program, mivel az ilyen tárgy nem csillog.

Minden tárgynak van felületi érdessége. Ezt az Imagine-ben a **Roughness** értékével állíthatjuk be. A nulla érték adja a legsímább tárgyat. Magasabb értékekkel hozhatunk létre durva felületű tárgyakat, mint például a kőszobor.

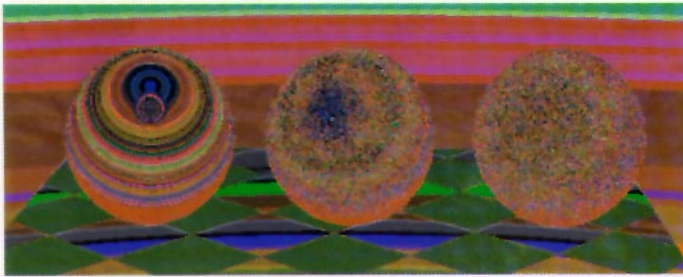
Fémes kinézetű tárgyak létrehozását az Imagine-ben még egy attribútum segíti, a **Shininess**. Amikor ez 0, a Filter és a Reflect az előbb elmondott módon működik. Ha a Shininess-nek nem 0 értéket adunk, akkor megváltozik a helyzet. A fényvisszaverés mértékét ez fogja meghatározni, nem a Reflect.

Egy fontos különbség azonban van a kétféle tükröződés között: Míg a Reflect tökéletes tükröződést ad, addig a Shininess elmosódott, diffúz tükröződést. Felismerhetők a tárgyak, de nincsenek éles kontúrok.

Ha **Shininess**-t használunk, a **Filter** nem az átlátzóságot szabályozza, hanem a tárgy csillogó színét. Például egy fémesen csillogó aranyérmének nem magas Reflect-et kell adni, hanem sárga Filter-t és nagy Shininess-t. Magas Reflect-el, 0 Shininess-szel tükör keletkezik.

Figyelem! A Shininess-szel együtt használt Filter

Attribútumok



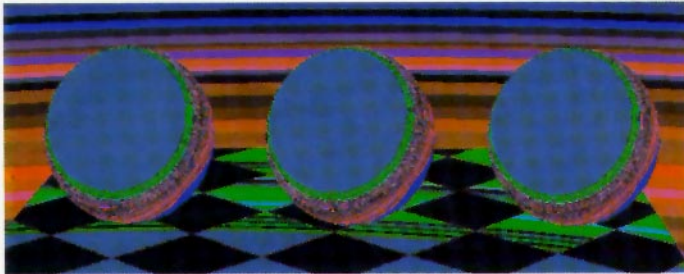
Rough-
ness

0
120
255

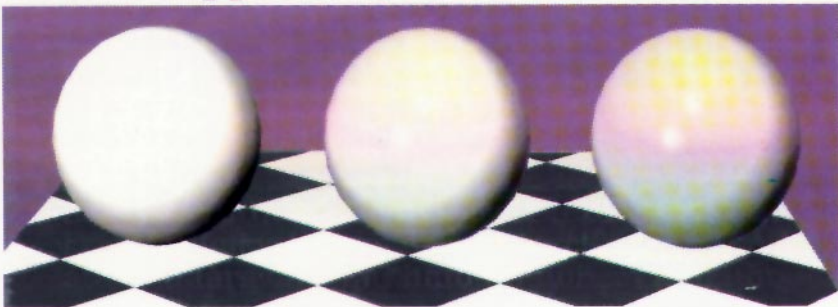
A Roughness-szel a tárgyak felületi érdességét állíthatjuk be. Ez nem módosítja a tárgy valódi alakját, csak a kinézetét.

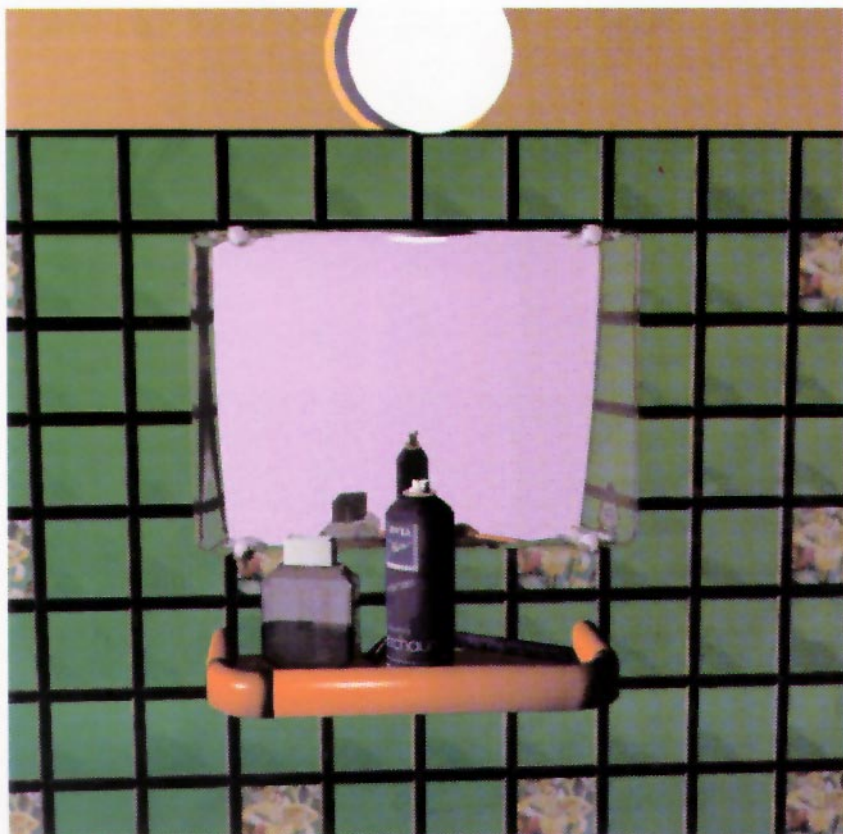
Shini-
ness

0
120
255



A tárgyak Shininess paraméterével azok fémes megjelenését tudjuk szabályozni. Ha az Index of Refraction nulla (mint a fenti képen), nem sok jelentősége van. A fénytörési mutató növelésével a tárgy fényszórása csökken. Az alsó képen a Shininess végig 255, míg az IofR 1, 2, és 3.55.





Ezen a képen brushráfeszítés hegyek vannak. A csempét három brush alakította ki egy síklapból. A fugázást egy altitude mappal, a csempék színét és a virágmintát egy 24 bites color mappal, míg a fényvisszaverőképességét egy reflect mappal készítettem el. A borotvahabon és az arcszeszen egy-egy color map hozta létre a címkéket. A borotva nyelén is van brush, bár nem igazán látszik, egy altitude map adja meg a recézést.

nem bírálja felül a Color-t, nem úgy mint a Reflect, így annak összhangban kell lenni a Filter-rel beállított csillogáshatással!

Hasonlóan a Reflect-hez, a Shininess is csak trace módban működik teljesen.

A Shininess-szel kapcsolatban áll az **Index of Refraction** input mező értéke. Ez szabályozza a felület fényszórásának mértékét. Ha az I of R 1, akkor a legjobban elmosódott a Shininess-szelt tárgyon a tükörkép. Az I of R értékét növelve a kontúrok élesíthetők, de még az igen magasnak számító 3-as érték sem hozza azt, mint a Reflect tükröződése.

Textúrák

A harmadik attribútum csoportba tartoznak a textúrák, amelyeket a **Textures 1-4** kapcsolókkal tudunk aktiválni.

Mielőtt ezek ismertetéséhez fognánk, képzeljük el a következő feladatot. Készíteni kell egy házfalat, amely látszólag téglából épült. Erre több megoldás is létezik az Imagine-ben. Az egyik, hogy úgy osztjuk fel a fal felületét, hogy az egyes felületelemeknek önálló színezésével kialakuljon a téglamintázata. (Azt, hogy minden téglát külön tárgyként, nem is említem) Elképzelted, hogy ez nem egy haladós módszer.

Másik megoldás, hogy egy bitmap rajzoló programmal, például a DPaint-tal rajzolunk egy téglamintázatot, majd ezt a képet a megtervezett falfelületre feszítjük, hogy módosítsa annak színeit, azaz a téglamintázat látszódjon rajta. A legtöbb ray-tracer program ezt a megoldást használja. Előnye, hogy nem csak téglát, de bármilyen mintázatot is létrehozható a felületen. Hátránya, hogy a brush-t (ez a

Detail editor

bitmap kép megnevezése a szaknyelven) mindíg meg kell rajzolni, ráadásul használata jelentős memória mennyiséget emészt fel és a rendering is lassul.

A brush-ráfesztéses eljárást az Imagine is tudja használni, erre szolgálnak a negyedik oszlop kapcsolói.

Van azonban egy harmadik módszer is, amely az Imagine egyik sajátossága. Bizonyos gyakoribb mintázatokhoz elkészítését automatizálni lehet. Ezek az ún. **Textúrák**, amelyek a már említett harmadik oszlop kapcsolóival aktiválhatók.

A példánál maradva, téglafalat létrehozhatunk úgy is, hogy egy erre a célra megírt Textúrában megadjuk a téglák és a habarcs méretét, elhelyezkedését, színét, majd a program a rendering alatt elkészíti ezekből az adatokból a téglamintázatú falat.

A textúra nem csak a tárgy felületén jön létre, hanem annak belsejében is, akár a teljes keresztmetszetben, ennek ellenére nem módosítja a tárgy fizikai szerkezetét.

A textúrák dinamikusak, paramétereik változtatásával különféle mintázatok hozhatók létre. A tárgyra helyezett textúra később, a tárgy méretének változtatása során is megőrzi eredeti arányai. Például, ha a téglafal 20 sor téglából állt, a fal kétszeresére történő nagyítása után is 20 sor téglából fog állni. Óriási előnyük, hogy nem igényelnek nagy memóriát és alig lassítják a képszámítást.

Az Imagine nem csak a téglafalra, hanem több különböző mintázatra is tartalmaz textúrát. Ezek a textúrák, amelyek külön fájlokban vannak tárolva, általában a Textures tartalomjegyzékben, kibővíthetők, azaz megfelelő programozói ismeretekkel és dokumentációk birtokában újak is létrehozhatók.

Minden további általános ismertetés helyett vegyük

sorba a lehetőségeket. Az Imagine 2.0-s változatával gyárilag 14 textúrát kapunk. Ezekkel a következő mintázatokot tudjuk kialakítani:

- Angular** - Színátmenet térben
- Bricks** - Téglaminta
- Camo** - Álcázóminta
- Checks** - Kockás
- Checks2** - Kockás
- Disturbed** - Zavaros, érdes
- Dots** - Pöttyös
- Grid** - Rácsos
- Linear** - Színátmenet egyenes mentén
- Pastella** - Pasztell
- Radial** - Színátmenet körkörösén
- Spots** - Foltos
- Waves** - Hullámos
- Wood** - Fa

Ahhoz, hogy textúrát adjunk a tárgyra (egyszerre négy lehet rajta), rá kell klikkelni valamelyik Texture kapcsolóra. Egy fájl kérdező jelenik meg, ahol a külső textúra fájlt kell megadni. Mint mondtam, ezek általában a Textures könyvtárban vannak. Ha egyszerre több textúra van a tárgyon, a prioritásuk annak megfelelő, ahányadik helyen vannak. Például a Texture 2 textúrája átfedés esetén letakarja a Texture 1-et.

Legyen az első textúra amelyet kipróbálunk, az **Angular**. Ezzel gyönyörű, térbeli színátmeneteket hozhatunk létre a tárgyakon. Miután kiválasztottuk a textúrát, egy nagyméretű, sok input mezőt tartalmazó kérdezőt kapunk. Mivel minden textúra kérdezője, a textúraparaméterek kivételével azonos, bizonyos dolgokat csak itt írok le, más textúránál legfeljebb csak utalok ezekre.

Az első input mező a **Texture Filename**, a textúra

Detail editor

nevét tartalmazza elérési úttal együtt. Mivel a program a saját könyvtárát tekinti alap könyvtárnak, így az azon belül lévő Textures fiókot nem kell abszolút úttal együtt megadni, elég a relatív útmeghatározás.

Mások által készített objektok felhasználásakor gyakran kaphatunk rendering közben "**texture not found**" hibaüzenetet. Ez azért is lehet, mert annak a tárgynak más elérési útvonalon van a textúrája, de az is lehet, hogy saját fejlesztésűt használ. Ilyenkor ez az input mező segíthet megtalálni a hiba okát.

Az **Apply to Children** kapcsolóval állíthatjuk be, hogy a textúra ne csak az adott tárgyra, hanem a belőle származó csoporttagokra is kiterjedjen.

A **Restrict to Subgroup** mező arra szolgál, hogy a textúra megjelenítését lekorlátozzuk valamely alcsoportra. Ide kell beírni a kívánt alcsoport nevét, és a textúra csak ezeken a felületelemeken jelenik meg.

Bizonyos textúráknál lényeges az elhelyezkedése, tengelyeinek iránya. Például a fa textúra száliránya, az Y tengellyel párhuzamos, évgyűrűinek közép-pontja a textúra Y tengelyén helyezkedik el.

A textúrát pozicionálni kétféle módon lehet, csakúgy mint a tárgyakat: interaktívan és számszerűen. Előbbihez az **Edit Axes** kapcsolóra kell klikkelni, utóbbihoz a **Transform Axes**-re. Mivel mindkét transzformációs módot ismerjük már, ezért itt most nem részletezem ezeket. Egyetlen különbség, hogy a **Transform Axes Only** nem használható.

A textúrák elhelyezésekor tudnunk kell, hogy általában a textúra X-Z síkja jelenti a felületi elhelyezkedését, Y pedig a mélységét. Például, ha a vastagsággal is rendelkező téglafalat teljes egészében beteríti a textúra, de az Y tengely csak a fal közepéig

ér, a fal túloldala nem lesz téglás. Ha erre mégis szükség volna, a textúrának át kell érnie a tárgyat.

A **Drop** kapcsolóra klikkelve dobhatjuk el a szerkesztett textúrát, vagyis törölhetjük a tárgyról.

Az idáig leírtak minden textúránál azonosak, most következnek az egyéni lehetőségek. Ha jól emlékszem, az Angular-nál tartottunk.

Az **Angular** három dimenziós színátmenetet eredményez a tárgyakon. A színátmenetek a tengelyek mentén jönnek létre. A kérdezőben öt irány végső színeit adhatjuk meg, az X tengely pozitív, az Y pozitív és negatív, valamint a Z tengely pozitív és negatív irányába esőket. Az X tengely negatív irányába eső szín a tárgy **Color**-nál beállított színe. (Más textúráknál is előfordulhat, hogy több szín meghatározásából az egyik a tárgy saját színe.)

Az egyes irányok színeit az input mezőkbe írt színösszetevő értékekkel adhatjuk meg, a szokásos módon. A bal alsó sarokban lévő négyzeten ellenőrizhetjük a beállított színeket.

További magyarázat helyett álljon itt egy kép. A textúra beállításait alapon hagytam, a tárgy színe 255,255,0.

ABC rendben a következő textúra a már emlegetett téglá, angolul **Bricks**.

Az **X,Y,Z Size** a téglá mérete, a **Mortar Size** a téglák között lévő habarcs vastagsága.

Rendes téglafalban a téglák kötésre állnak, általában minden téglá fél szélességgel, vagy hosszal van arrébb mint a másik. Ezt is modellezni tudjuk az Imagine Bricks textúrájában. Az **X Shift With Z** a téglasorok X irányú eltolását jelenti, ez általában fele szokott lenni a téglák X méretének. Az **X Shift With Y** az X irányú eltolás Y függvényében. A **Z Shift With Y** pedig a téglák Z irányú eltolását jelentik.

Detail editor

A téglák színe, fényvisszaverő és fényáteresztő képessége a tárgynál beállítottal azonos, míg a habarcsra ezeket a Mortar-ral adhatjuk meg.

A harmadik textúra, amivel megismerkedünk, a **Camo**. Ez egy álcázómintát hoz létre a tárgyon öt szín véletlenszerű foltokban való elhelyezésével. A foltok méretét a **Spot Spacing**-gal tudjuk beállítani. A **Random Seed** a véletlengeneráláshoz kell. Ennek bármilyen értéket megadhatunk, ez lesz a véletlengenerátor kezdőértéke.

A Camo kérdezőjében négy szint határozhatunk meg, a szokásos módon. Az ötödik szín a tárgy saját színe. Az egyes színfoltoknak azonos a fényvisszaverő és fényáteresztő képessége, a tárgynál beállított érték szerint.

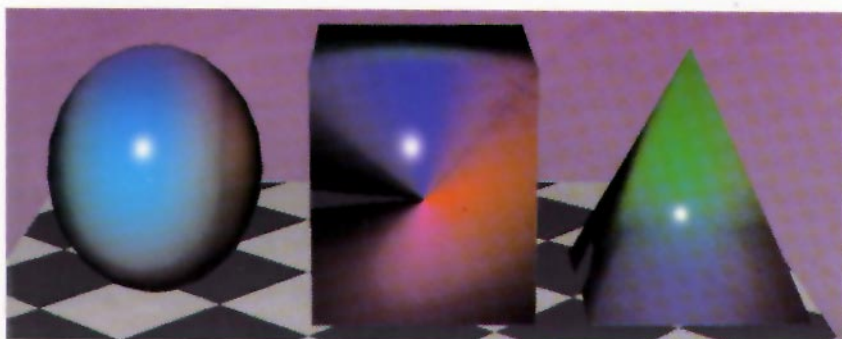
Nem föltétlen szükséges egyszerre minden szint felhasználni a **# of Colors** mezőbe kerül ezeknek a száma. Itt minden esetben a Camo színeinek számát kell megadni, tehát ehhez hozzájön még a tárgy saját színe.

A **Spot Area** mezőbe a foltok által lefedett felületek aránya kerül. Ennek értéke 0-tól 1-ig változhat. A 0 érték azt jelenti, hogy a tárgyon nem jönnek létre foltok. Ha a Spot Area 1, a tárgy eredeti színe nem látszik, mivel azé a maradék terület.

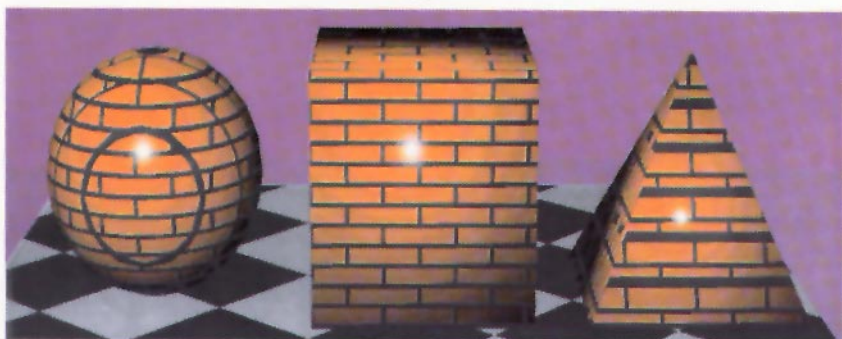
Kockás textúra kettő is van, **Checks** és **Checks 2** néven. Ezek között egyetlen különbség, hogy a Checks a Z tengely irányában is létrehozza a kockákra osztást, míg a Checks2 e tengely mentén nem.

A kockák méretét a **Check Size** input mezőben kell megadni. Ezek mérete minden irányban azonos.

A kockák színét, fényvisszaverő és fényáteresztő képességét a **Color**, **Reflect** és **Filter RGB** mezők-



Angular

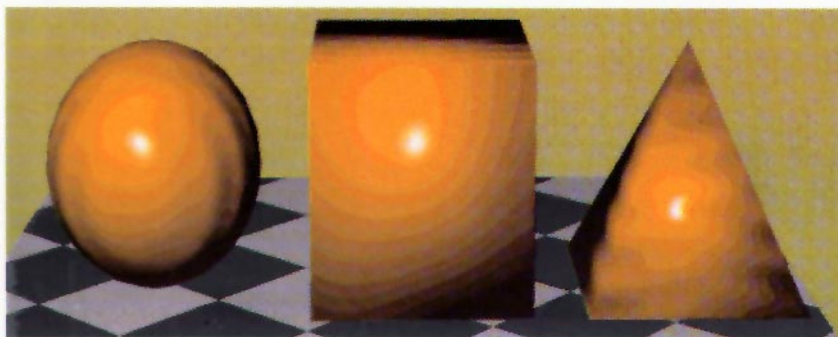


Bricks

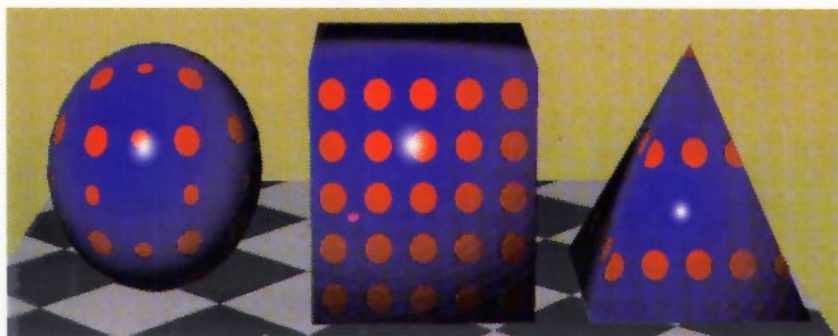


Camo

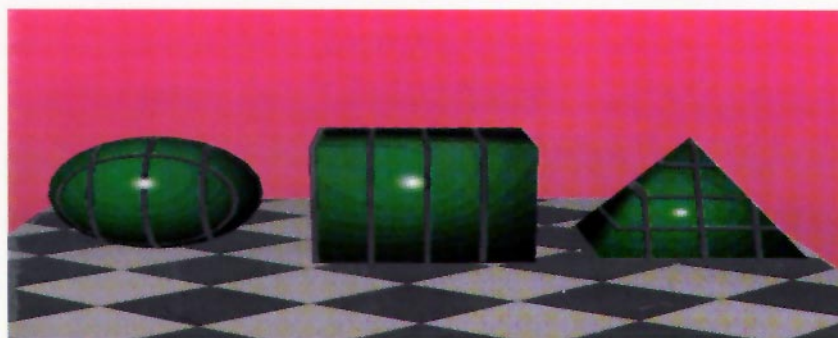
Detail editor



Disturbed



Dots



Grid

ben a szokásos módon határozzuk meg. A kockás minta másik színét a tárgy saját színe adja.

A **Disturbed** textúrával egymást keresztező hullámvonalak segítségével felbolygatott, zavaros felületet hozhatunk létre. Ehhez nem kell új színt megadni, a textúra a tárgy saját színének alkalmazásával működik. Nagyon jól használható például vízfelület készítésére.

Az **Amount** a hullámok amplitúdója. Ha 0, nincs hatása. Értéke kisebb kell legyen 1-nél.

A **Wavelength** a hullámhossz. Minél kisebb, annál több hullám jön létre a tárgyon. Sem a túl kicsi, sem a túl nagy érték nem ad szép eredményt. Gyakorlatilag a tárgy hosszának $1/5$ - $1/20$ -a a megfelelő.

Az **X Separation** a hullámforrások középpontjának távolsága. Ha ez 0, nem jön létre interferencia. Általában a hullámhosszal közel azonos értéket szoktunk megadni, ez vált be legjobban.

A **Small < 1** értékének szintén kisebbnek kell lenni mint 1. Segítségével kis változatosságot adhatunk a mintázatnak.

Sorban a következő textúra a **Dots**, amivel pöttyöket hozhatunk létre. A pöttyök sűrűségére a **Dot Spacing** van hatással, ami a szomszédos pöttyök középpontjainak a távolságát jelenti.

A **Dot Radius** a pöttyök sugara. A pötty nem sík korongból áll, hanem egy gömbből. Ezért van, hogy görbe tárgyakon vagy ferde felületeken néhány pötty átmérője kisebb a többinél. A túl nagy pötty az egész tárgyat eltakarhatja!

A pöttyök színét, fényvisszaverő és fényáteresztő képességét a **Color**, **Reflect** és **Filter RGB** mezőkben adhatjuk meg.

A **Grid** textúra rács mintázatot hoz létre. A rács

Detail editor

sűrűsége a **Grid Size** input mezőben, a rács vonalainak vastagsága pedig a **Line Size**-ben adható meg.

A rács vonalainak színét, fényvisszaverő és fényáteresztő képességét a **Color**, **Reflect** és **Filter RGB** mezőkben kell megadni.

A **Linear**-ral színátmenetet hozhatunk létre a tárgyon. A színátmenetben nem csak a tárgy színe, hanem a fényvisszaverő vagy fényáteresztő képessége is változhat. Az átmenet a Z tengely mentén jön létre, a textúra tengelypontjától kezdődően. A kiinduló szint, fényvisszaverést és fényáteresztést a tárgy színe adja, a végsőt pedig az itt megadott **Color**, **Reflect** és **Filter**. Azt, hogy a színátmenet milyen élesen, mekkora távolságon jön létre, a **Transition Z With** szabályozza. Ilyen szélességben keverednek a színek. Ha ez nagyobb, mint a tárgy szélessége, az átmenet nem fogja elérni a végső szint.

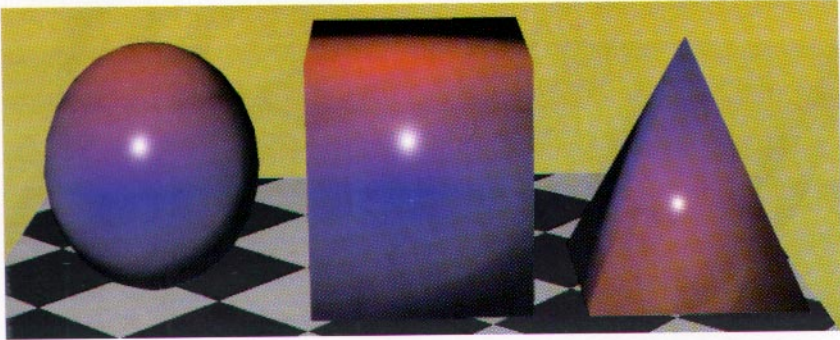
A **Pastella** textúra hasonló a Camo-hoz, a különbség az, hogy a foltok nem rendelkeznek éles kontúrral, hanem lágy átmenettel érintkeznek. A foltok betérítik a tárgy teljes felületét.

A **Detail Size** az egyes foltok átlagos mérete. A **Random Seed** a szokásos véletlenszám. **Color**, **Reflect**, **Filter** a már ismert funkciókat töltik be.

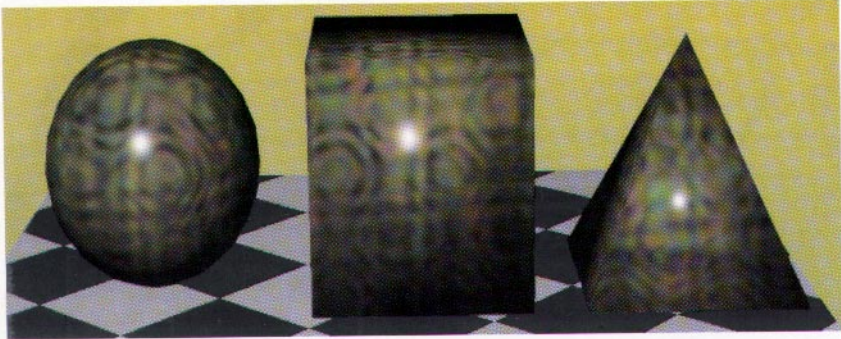
A **Radial** textúra a Linear-hoz hasonlóan színátmenetet készít, a különbség, hogy nem egyenes mentén, hanem koncentrikusan. A színátmenet a textúra tengelypontjából indul.

A **Start Radius** által meghatározott sugáron belül a tárgy eredeti színe van jelen, ezen túl kezdődik a színátmenet. Az átmenet hosszát a **Transition Z With** szabályozza.

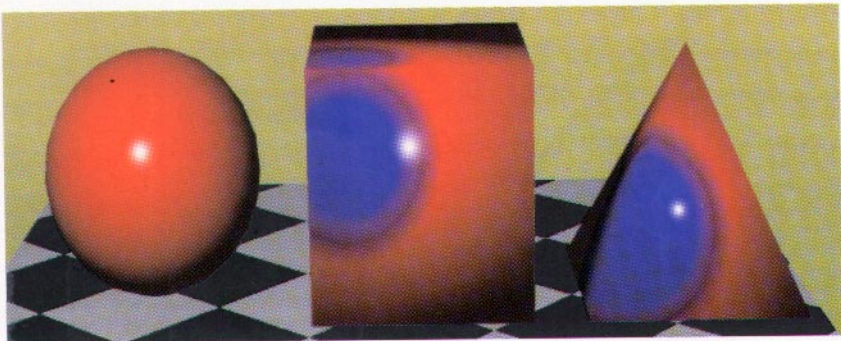
A gömbszerű színátmenet miatt a Radial tetúra



Linear

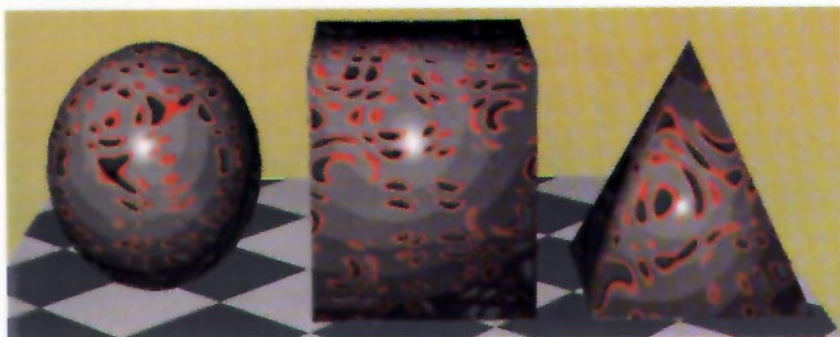


Pastella

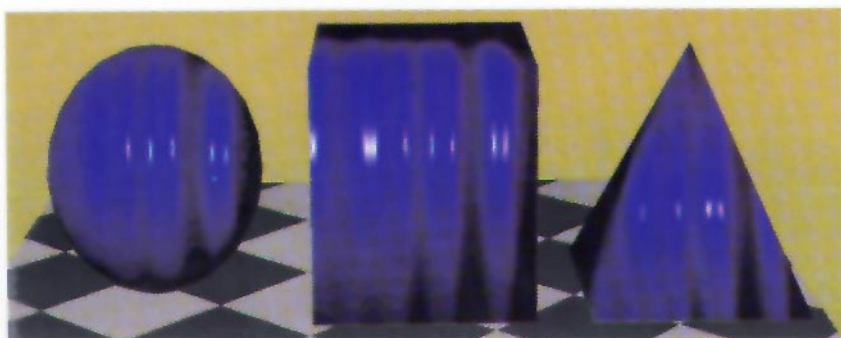


Radial

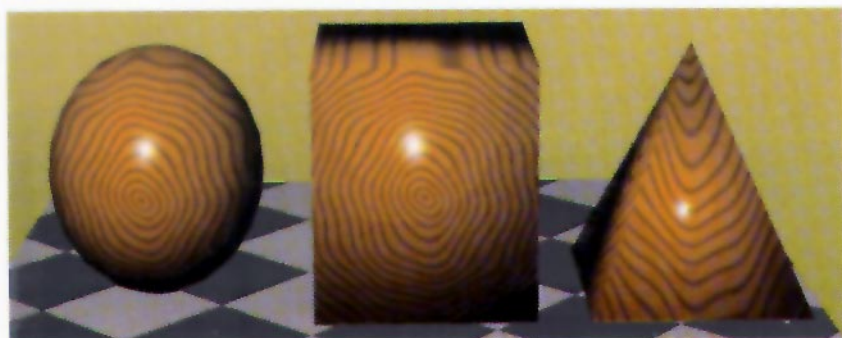
Detail editor



Spots



Waves



Wood

gömbön nem látszik.

A **Spots** egy nagyon látványos díszítő textúra, amellyel érdekes, alakatlan foltokat hozhatunk létre a tárgyakon.

A **Spot Spacing** a foltok távolsága, a **Random Seed** pedig a szokásos véletlenszám.

A textúra a tárgy színén kívül két szint igényel. A **Color#1** a foltok körvonalának színe, a **Color#2** pedig a foltok belsejéé. Legjobb eredményeket a kontrasztos, jól elkülönülő színek adják.

A **Waves** hullámzást hoz létre a tárgyon. Hasonlóan a **Disturbed**-hez, ez sem hoz létre új színeket, csak a meglévő színen hoz létre hullámzó látványt. A hullámzás itt egy pontból indul, nincs tehát interferencia az egyes hullámok között.

A **Wavelength** a hullámhossz, az **Amplitude** pedig a hullámok magassága.

A **Distance Traveled** a hullámzás távolsága. Ennek változtatásával hullámzó mozgás készíthető az animációban.

Elérkeztünk az utolsó textúrához, a **Wood**-hoz. Ezzel fa erezetű tárgyakat készíthetünk. Az évgyűrűk a textúra X-Z síkján helyezkednek el, a fa száliránya az Y tengellyel párhuzamos. A fa színe a tárgyéval azonos, míg az évgyűrűkét a **Color** input mezőkben adhatjuk meg.

Az évgyűrűk közti távolságot a **Ring Spacing**-gel lehet beállítani, míg a gyűrűk szélességét az **Exponent**-tel.

A **Variation1** az évgyűrűk szabályosságát befolyásolja. A 0.1 körüli értékkel szabályos évgyűrűket alkothatunk, míg nagyobbakkal kuszábbakat.

A fa textúra elkészítéséhez is véletlenszám gene-

Detail editor

rátort használ a program. Ennek kezdőértékét a **Random Seed**-be lehet beírni.

A Wood-dal végére értünk az Imagine textúrák ismertetésének. Általánosságban elmondható minden textúráról, hogy paramétereik változtatásával animációkban is felhasználhatók, például folyamatos hullámzást hozhatunk létre, vagy egy rácsos tárgyat átalakíthatunk kockássá, stb.

Bizonyos textúráknál nem csak Color, hanem Reflect és Filter értékeket is megadhatunk. Ha a tárgyra beállított Shininess nem 0, ezekre is ugyan az a szabály vonatkozik, mint a tárgy Reflect és Filter értékeire, vagyis a Reflect nem számít, és a Filter-től függ a diffúz tükröződés színe.

A textúrák egymással kombinálhatók, például egy Angular-ral színezett tárgyat hullámoztathatunk a Wavews-szel. A Restrict to Subgroup-pal megoldható, hogy egy fa keretben Spots-szal készül absztrakt kép legyen, stb.

Ha a textúrák átfedik egymást, a nagyobb sorszámú takarja a kisebbet.

Kísérletezéssel sok érdekes lehetőséget tapasztalhatunk ki, például a fa megfelelő színek és egyéb paraméterek segítségével márvány, vagy más hasonló mintázat kialakításához is használható, stb.

Brush-ok

Az imént ismertetett textúrák nem az egyedüli lehetőségek, hogy tárgyainkon mintázatokat hozunk létre. Ha ezek között nem találnánk megfelelőt, normál IFF-ILBM (PC-n TIFF) képeket is elhelyezhetünk tárgyainkon. Ezek előnye, hogy bármilyen mintázatok létrehozhatók, hátrányuk viszont, hogy

nagy mennyiségű memóriát kötnek le számítás közben. Hátrányuk ellenére mindenképpen szükség van rájuk.

IFF képet (továbbiakban brush-t) a tárgyra feszíteni a Brush 1-4 kapcsolók valamelyikére klikkelve lehet. Egyszere négy brush lehet a tárgyon, prioritásuk a Brush # -al azonos. A Brush 2 átfedés esetén letakarj a Brush 1-et. A brush-ok bármilyen méretű és felbontású szabvány IFF/ILBM (mint említettem, PC-n TIFF) képek lehetnek a két színűtől kezdve a 16.8 millió színűig.

A kapcsolóra klikkelve megjelenik egy fájl szelektor, ahol kiválaszthatjuk a kívánt képet. Ha ez megtörtént, újabb kérdező nyílik, ahol a brush-okkal kapcsolatos beállításokat tehetjük meg.

A kérdező tetején, a **Filename** input mezőben van a brush neve és elérési útvonala, amit bármikor megváltoztathatunk.

Kicsit lentebb a **Type** alatt négy kapcsolót találunk, amikkel a brush felhasználásának típusát állíthatjuk be. Melyek lehetnek ezek?

A tárgyra feszített brush megváltoztathatja annak színeit, vagyis azon a helyen a tárgy a ráfeszített kép színét, mintázatát veszi fel. Ez a **Color mapping**.

A kép nem csak a tárgy színére lehet hatással, hanem a fényvisszaverő képességre is. Minél világosabb a brush, annál nagyobb lesz ott a tárgy fényvisszaverő képessége. A fekete részeken nincs tükröződés, míg a fehéren ez maximális. Természetesen a Reflect map lehet színes is, a piros kép a piros szín visszaverését eredményezi, csakúgy mint a Reflect RGB beállítása. Gondolom mindenki rájött, hogy ezt a módot a **Reflect** kapcsolóval választhatjuk ki.

A **Filter** hasonló hatású, de a fényáteresztő képes-

Detail editor

séget tudjuk a brush-sal szabályozni. A fekete képnél átlátszatlan a tárgy, míg a fehérenél teljesen átlátszó. Most is használhatunk színes brush-t, hatása értelemszerű.

A legérdekesebb mapping az **Altitude**. Ezzel a tárgy felületi kialakítását, "domborzatát" tudjuk szabályozni. A domborzat a ráfeszített kép világosságától függ, a világosabb helyeken magasabbra emelkedik a felület. Ez a tárgy színére nincs hatással, ezért elegendő a 16 vagy 256 szürke árnyalatú kép használata. Lehet ugyan a brush színes, de akkor is a pixelek fényessége számít.

A brush beállításai alap esetben nem vonatkoznak a tárgy gyermekeire. Ha azt akarjuk, hogy a tárgy leszármazottjaira is vonatkozzon, az **Apply to Children** kapcsolót kell kiikszelni, hasonlóan mint a textúráknál.

Most is van lehetőség a képnek egy bizonyos felületrészre való lekorlátozására, az alcsoport nevét kell a **Subgroup** input mezőbe írni.

Hogyan kerül a kép a test felszínére? A legegyszerűbb eset, hogy párhuzamos vetítéssel, hasonlóan mint ahogy a vetítógép a vászonra vetíti a képet. A vetítés síkja a X-Z sík. Ehhez a Method kapcsolói közül a **Flat X**-et és a **Flat Z**-t kell bekapcsolni. A kapcsolók azt jelentik, hogy mind az X, mind a Z tengelye a képnek egyenesre kerül.

Az Imagine automatikusan elhelyezi úgy a brush-t, hogy az az egész tárgyat befedje, de ezen lehet változtatni. Klikkelj az **Edit Axes** kapcsolóra. Hasonló sárga befogadó keretet kapsz, mint a textúráknál. Az elől nézet keretének jobb felső negyedében van a tárgy. Síkvetítési módban a brush a befoglaló keretnek ebben az X-Z tengelyekkel határolt negyedében van, erre figyelj oda!

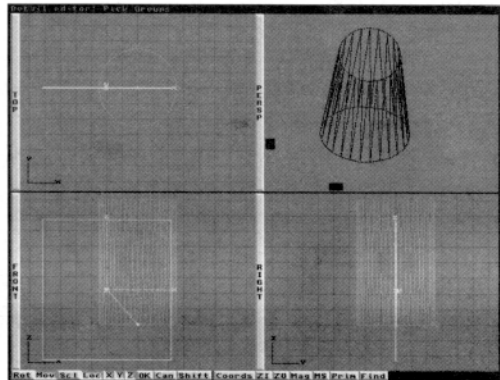
A tárgyra feszített brush minden esetben kétdimen-

ziós, X-Z sík, miért van mégis Y irányú kiterjedése a bounding boxokban, az oldalt és felülnézetben?

Ez az Imagine vetítési rendszerének sajátossága, nem csak a tárgy felszínére tud vetíteni, hanem annak mélységében is. Azt, hogy egy tárgyban milyen mélyre hatoljanak a vetítő sugarak, a brush Y tengelyének kiterjedése adja meg. Ha ez átéri a tárgyat, a brush a tárgy túoldalán is látszik, persze fordított képállásban.

Az idáig elmondottak a Flat X-Flat Z síkvetítésre vonatkoztak. Mi a helyzet, ha a képet nem sík, hanem összetett felületre akarjuk felfeszíteni? Próbálkozhatunk az előző síkvetítéssel, de a görbe felületeken a kép torzulni fog. Tegyük fel, hogy egy palackra szeretnénk címkét tenni. A legjobb megoldás az volna, ha a címkét körben tudnánk a palackra feszíteni. Nos, az Imagine-ben van erre mód. Legyen a palack a front nézetben álló helyzetű, ekkor a címke a Z tengely körül hengeredik, míg X irányban egyenes.

Az ábrán a palackot egy Tube primitív helyettesíti. A kívánt feszítési módot a **Method** kapcsolóival állíthatjuk be. A **Flat X**-et meg kell hagyni, míg a **Flat Z**-t át kell váltani **Wrap Z**-re. Ez jelenti a Z körüli csavarást. Ebben az esetben a brush-t némiképp máshogy kell elhelyezni a tárgyon, mint a **Flat X** - **Flat Z** vetítésnél. A vetítés a brush tengelypontjából történik, a Z tengely körül, az X-szel párhuzamosan. A brush Y



Detail editor

méretének minél kisebbnek kell lenni. Mivel a vetítés egy pontból körkörösén történik, az X méreteinek nincs jelentősége. A vetített kép magasságát a Z tengely hossza határozza meg. Ha a brush nem a tárgy középpontjában van, hanem annak valamely széle felé, a brush összenyomódik azon a részen, amihez közelebb van, és széthúzódik a tárvolabbi felületen, csakúgy mintha valóban vetítenénk.

A vetítés nem csak a Z tengely, hanem az X körül is végbemehet. Ezt a **Wrap X**, **Flat Z** kapcsolással érhetjük el. Az előbb elmondottak érvényesek ekkor is, azzal a különbséggel, hogy az X és a Z szerepe felcserélődik.

Az X és Z tengely körüli hengeres vetítést egyszerre bekapcsolva, gömbre vetítést hozhatunk létre. **Wrap X**, **Wrap Z** módban is a hengeres vetítéskor elmondottak érvényesek a brush-re, de sem a Z, sem az X tengelyek méretének nincs jelentősége.

Ismétlődő mintázatok kialakítását két funkció segíti. A **Repeat** bekapcsolásával a brush hézag nélkül ismétlődik a test teljes felületén, az egyes másolatok összeérnek. Síkvetítéskor az ismétlődés X és Z irányban egyaránt létrejön, hengeres vetítéskor csak a Flat tengely mentén, míg WrapX - Wrap Z vetítéskor nincs ismétlődés. A kiindulási brush méretét ugyan úgy kell meghatározni, mint Repeat nélkül.

A **Mirror** bekapcsolásával az ismétlődő brush-ok mindig tükröződnek az ismétlődés irányára merőlegesen. Ezzel el tudjuk érni, hogy az ismételt képek látható törés nélkül illeszkedjenek össze, mivel a kép mindig a tükörképével illeszkedi.

A **Repeat** funkció megkönnyíti olyan felületek létrehozását mint a csempfal vagy sormintás felület.

Az **Inverse Video**-val negatívba fordíthatjuk a



Egyik népszerű téma a 3D modellezésben a földgömb. Ezt létrehozni igen egyszerű, egy világterképet kell Wrap X, Wrap Z eljárással gömbre feszíteni. Nem mindegy azonban, hogy milyen térképet alkalmazunk. A szokásos kúpvetületű és Mercator féle térképek az eljárás során torzulni fognak. Ez a torzulás az amerikai részen a legszembetűnőbb. Ennek elkerülésére hengervetületű térképet kell alkalmazni. Ilyeneket használnak az időzóna és a meteorológiai térképeken. Itt a szöveg alatt is egy ilyet rejtettem el.

Nagyon szép földgömböt lehet létrehozni, ha a világterképet nem csak color, hanem altitude mapként is felhelyezzük a gömbre. Az altitude map-et általában át kell szímezni, a magasságoknak megfelelően.



A televízióban nem csak természetfilmeket adnak, hanem koncertet is. A 220. oldalon a Galériában maga Beethoven zongorázik a mellettünk lévő hangszeren. Ha nem hiszed, lapozz!

tárgyra kerülő képet, látványos effektust hozva létre.

A **Use Genlock** videós alkalmazáskor hasznos. Kikészelve a kapcsolót, olyan kép készül, amit ha genlockkal video jelre keverünk, a brush háttér színének helyén a video jel látszi.

Az Imagine képes arra, hogy a tárgyra kerülő brush-t animálja is. Ehhez nem a DPaint-ban megszokott Anim Brush-ra van szükség, hanem önálló képkockákra, amiket azonos könyvtárban helyezünk el. A képeket azonos néven, a 0001, 0002, 0003, stb kiterjeszésekkel kell ellátni. Amígán a kiterjesznek mindig 4 karakteresnek kell lenni, PC-n a DOS korlátai miatt csak 3-nak. A brush nevének megadásakor a kiterjesztést és a pontot nem kell beírni. A képkockák maximális száma kerül a **Max Sequence #** input mezőbe. Amikor az így elkészült tárgyat egy animációban felhasználjuk, minden kockán a következő brush kerül a tárgyra. Ha végére ért a brush-oknak, kezdi előről. Ezzel a módszerrel könnyedén készíthetünk működő televíziót, mozi vásznat, stb. Egy érdekes alkalmazása például TV-stúdiót ábrázoló animációban van, ahol a stúdió monitorára mindig az előző képkockát feszítjük brushként. Ezt könnyedén meg tudjuk oldani, mivel az animáció képkockáit Pic.0001, Pic.0002, stb néven menti ki az Imagine, tehát brus-ként a képek útvonaláról a Pic-et kell beírni, Max. Sequence#-nek pedig az animáció hosszát.

Általánosságban ennyi mondható el a brush mappingról, de érdemes a feladat bonyolultságából adódóan az **Altitude** map-ról külön is szólni. Amikor egy képet Altitude map-ként feszítünk egy tárgyra, akkor a képnek megfelelő domborzat jön létre a felszínen. A domborzat nem változtatja meg a tárgy valódi alakját, csak úgy látszik! Minél világó-

Detail editor

sabb a képrészlet (pixel) a hozzá tartozó kiemelkedés annál magasabb lesz. Ezt az eljárást szokás Bump mapping-nak is nevezni.

A felület egyenetlenségei lágy átmenetekkel jönnek létre, függetlenül az alkalmazott kép felbontásától. Azt, hogy a fehér színhez tartozó kiemelkedés milyen magas legyen, az Y tengely (a brush-é) határozza meg. Gyakorlatilag ez a tárgy átmérőjének néhány százaléka. Egy meteorkráterekkel erősen szabdalt bolygó felszínénél kb 10%-a a bolygó átmérőjének. Ennél nagyobb értékre ritkán van szükség.

A **Full Scale Value** szintén hatással van a domborzat magasságára. Ennek értéke adja meg, hogy mely szürke árnyalattól számít egy szín fehérnek. Például ha ide 128-at írsz, már a középszürke árnyalat is maximális kiemelkedést hoz létre.

Az Altitude mapping a legmemóriaigényesebb brush kezelés, sokszor a 8 MB-át memória is kevésnek bizonyul egy összetettebb képhez 1-2 bump mappal.

Mind a textúrákról, mind a brush-okról elmondható, hogy dinamikusan kötődnek a tárgyakhoz. Ez annyit jelent, hogy azzal együtt mozgathatók, méretezhetőek. Ha a tárgyat, amelyen textúra vagy brush van, a Scale-lel átméretezünk, a rajta lévő mintázat a tárgyal együtt növekszik vagy zsugorodik, megtartva a relatív méretét és pozícióját. Például egy sakktábla, amelyet 8*8-as kockás textúrával hoztunk létre, méretváltoztatás után is 8*8-as marad.

Az előbb elmondottak csak arányos méretváltásra érvényesek maradéktalanul. Ha a méretezést csak egy, vagy két tengely irányában hajtjuk végre, nem biztos hogy megmarad a mintázat relatív mérete és pozíciója.

Egyéb attribútumok

Egyéb attribútumok

Az Attributes legnagyobb részén ezzel túlvagyunk, az a néhány kapcsoló és input mező ami még hátra van, már könnyen megérthető!

Az **Index of Refraction**-ról már volt szó a Shininess-nél, igaz csak futólag. Értéke a tárgy fénytörési mutatóját adja. Jelentősége átlátszó vagy Shininess-es tárgyaknál van. Minimális értéke 1, maximális nincs korlátozva, de a 3 körüli érték már igen magasnak számít.

Shininess-es tárgyaknál a visszaverődő fény szórását állítja be, 1-nél a leginkább diffúz a visszavert fény. Átlátszó tárgyaknál azonos a fizikában tanulttal, tehát a levegőhöz viszonyított fénytörési mutató. A leggyakoribb értékek a következők:

Levegő 1.0
Víz 1.33
Alkohol 1.36
Kvarc 1.46
Üveg 1.52
Borostyán 1.55
Gyémánt 2.4

Az Imagine csak a **Trace** képszámítási módban veszi figyelembe a tárgyak fénytörését, de ekkor pld. egy domború üveg nagyítóként viselkedik.

Megjegyzendő, hogy a magasabb fénytörési mutatók jelentősen lassítják a képszámítást.

A **Fog Length** input mező értékének segítségével a tárgyat ködszerűvé, áttetszővé tehetjük, ez merőben más mint a normál filter, hiszen nincs fénytörés, a ködön keresztül elmosódnak a kontúrok, stb. Minél sűrűbb a köd, a tárgy annál jobban elnyeli a fényt, és elmossa a körvonalakat. A ködön áthaladó fény

Detail editor

amellett hogy gyengül, színét is megváltoztatja a köd színének megfelelően. A köd színét az eredeti tárgy színe határozza meg.

A köd sűrűségét a **Fog Length** határozza meg. Ha ez 0, a tárgy nem mutat ködszerű tulajdonságokat. Az érték növelésével egyre ritkább köd hozható létre. Pontos iránymutatást nem lehet adni, mivel a köd fényre gyarkorolt hatása függ attól is, hogy az milyen hosszan halad a ködben. A ködös rész hossza természetesen a tárgy átmérője. Gyakorlatilag közepes köd létrehozásához a Fog Length a tárgy méretével azonos, Imagine egységben.

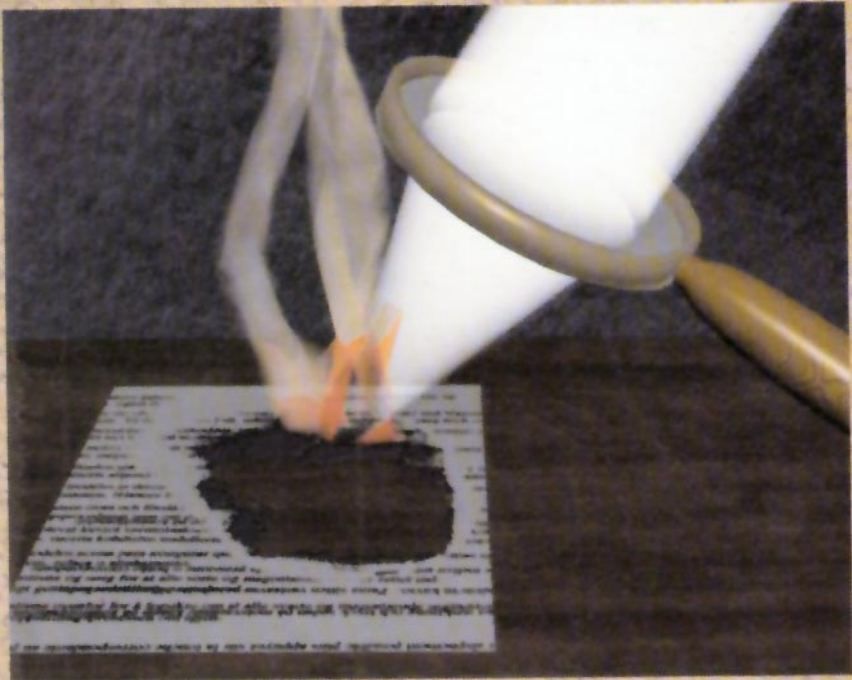
A köd attribútumú tárgyra nem vonatkozik sem a Reflect, sem a Filter. A textúrák és brush-ok (a Color és az Altitude) viszont rendesen kifejtik hatásukat, bár a köd kinézetű tárgyon a kisebb egyenetlenségek nem igazán látszanak. Az alig különböző színű foltokkal létrehozott Camo vagy Pastella textúra köd tárgyon nagyon élethű ködöt vagy füstöt eredményez. Ezeket a textúrákat animálva még gomolyoghat is a füst.

A köd a füst és felhők létrehozásán kívül eredményesen használható a fénysugár szimulálására is, mivel ez egyébként nem látható a képen.

Az Attributes kérdező alsó részén találunk öt kapcsolót. Ezekkel különböző hatású tárgyakat hozhatunk létre.

Az első kapcsoló a **Phong**. A könyv elején már tettem említést róla, ez felelős a lágyabb életmenetekért. Ha bekapcsoljuk, minden szögben található felületem között íves átmenetet próbál a program generálni, ellensúlyozva a poligonális ábrázolás szögleteségét.

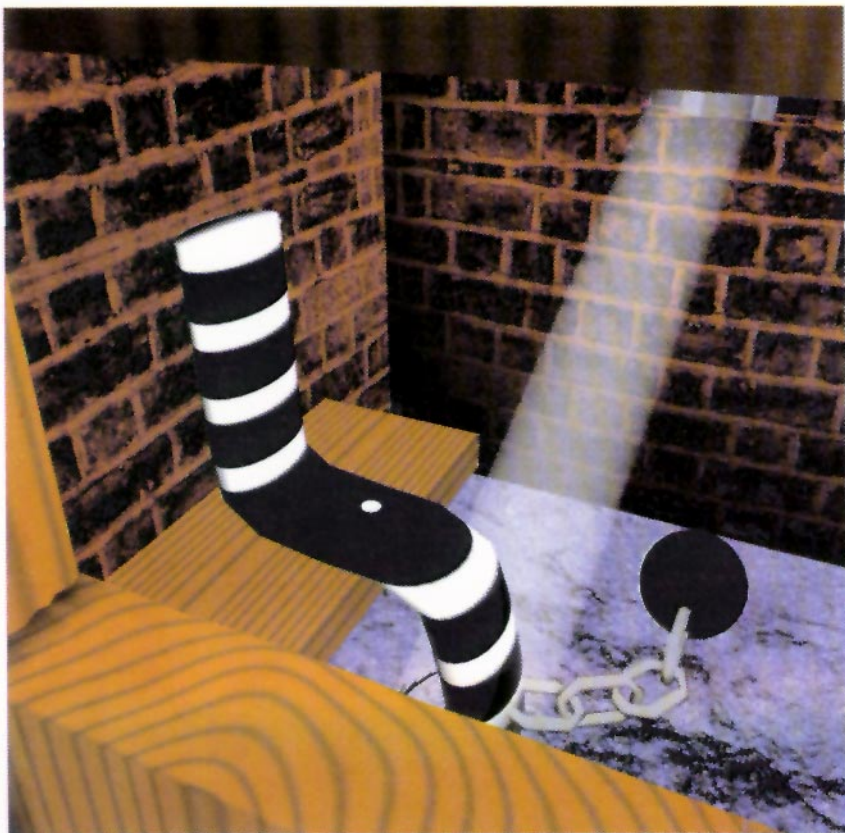
Nem ez az egyetlen módszer, hogy az éleket lekerekítsük, bár kétség kívül a legegyszerűbb. Hátránya,



A fénny útja útja nem látszik az Imagine által generált képeken, a fénny nem fókuszálható, tükrök segítségével nem irányítható. Ennek ellentmondani látszik a fenti kép. A kép elkészítéséhez fehér ködszerű tárgyat használtam, amelyet a fénycsővának megfelelőre alakítottam. A fényforrás maga a ködtárgy, úgy beállítva, mintha a fénysugár valóban abból eredne. Persze ez nem olyan, mint az igazi, de bizonyos fokú gyakorlattal, néhány próbálkozás után egész jól meg lehet közelíteni azt.

A láng és füst szintén köddel lett létrehozva.

A másik trükk, ahogy a papír átlátszik a kiegészítő darabon. Valójában az nem átlátszó, hanem az égett foltot kialakító brush-ba van beletéve ugyan az a kép, mint ami az asztalra került.



A fény útja nem látszik az Imagine által készített képeken, de egy trükkel létrehozhatunk ilyen hatást. A fénycsóva helyére egy megfelelő alakú ritka köd attribútumú tárgyat kell helyezni. A tárgyat nehéz úgy beállítani, hogy pontosan a fény helyére kerüljön, ezért célszerű magát a ködtárgyat használni lámpaként. Helyezd el a tárgy tengelypontját oda, ahonnan a fény áradni fog, az Y tengely pedig állítsd be a tárgy hosszában, a fény erre fog sugározni, a Light attribútum beállításai után.

Egyéb attribútumok

hogy csak ez egész tárgyra használható, márpedig sokszor kerülünk olyan helyzetbe, hogy ugyan azon tárgyon vannak lekerekítendő és élesen hagyandó élek is.

A megoldás, külön-külön kell minden élre meghatározni, hogy lágy vagy kemény legyen. Ez így még nehézkesnek tűnik, de mindjárt leegyszerűsítjük.

Pick Edges módban válaszd ki azokat az éleket, amelyeket nem akarsz tompítani, majd használd a **Functions/Make/Make Sharp** menüpontot. Ez annyit csinál, hogy jelzést rak a kiválasztott élekre: ezekre nem hatásos a továbbiakban a **Phong**. Ez a jelzés nem látható, ha később tudni akarod melyek a kemény élek, szintén **Pick Edges**-ben használd a **Pick/Select** menü **Pick Sharp** pontját. Azok az élek választódnak ki, amelyek keményítve vannak.

Az előzőekből az hiszem kitűnik, a tárgyra továbbra is alkalmazni kell a Phong-ot.

Ha valamelyik kemény élet újból lágyítani akarod, ki kell választani azt, majd **Functions/Make/Make Soft**-tal visszahelyezni a Phong hatása alá.

A közhiedelemmel ellentétben a Make Soft önmagában nem Phongolja az élet, csak megszünteti a keményítést, vagyis azt, hogy a Phong nem vonatkozik rá.

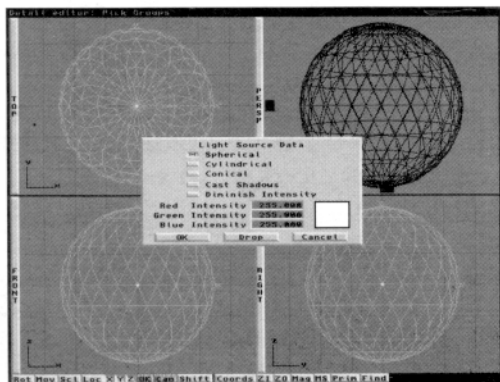
A **Fastdraw** kapcsoló mindössze annyit tesz, hogy a három síknézeti ablakban a tárgy részletes rajza helyett a befoglaló keretével ábrázolja azt, ugyan úgy mint a **Quickdraw**. Ezt a rajzolási módot vagy a kapcsoló kikapcsolásával, vagy a **Quickdraw None**-nal szüntethetjük meg. Utóbbi parancs a szerkesztőben lévő összes tárgy **Fastdraw** attribútumát kikapcsolja.

A **Bright** kapcsolót bekapcsolva a tárgyat nem árnyalja a program, hanem kizárólag a Color-nál beállított színnel rajzolja meg. Normál esetben a

Detail editor

tárgy látható színe az eredeti színén kívül a megvilágítás erősségétől és színétől is függ, a kapcsoló bekapcsolása ezt szünteti meg. Főleg világító felületek, mint például lámpa vagy nap, készítésénél használjuk, de egyéb tárgyakon is alkalmazhatjuk, rajzfilmszerű hatást hozva létre.

Tárgyaink fényforrásként is viselkedhetnek, ezt a **Light** kapcsoló kiiktetésével érhetjük el. Egy fényforrásnak többféle paramétere van, ezeket a kapcsolóra klikkelés után megjelenő kérdezőben állíthatjuk be.



A fény mindig a tárgy tengelypontjából sugárzik, függetlenül a tárgy kiterjedésétől. Tehát például neoncső készítésre nincs mód közvetlenül. A **B r i g h t - t a l** kombinálva már elérhetünk hasonló hatást, a cső teljes

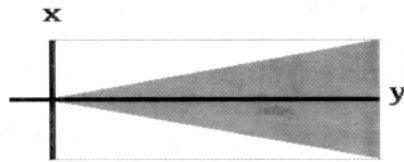
méretében fényes lesz, de a fény ekkor is tengelypontból ered.

A tárgy három féleképpen sugározhatja a fényt, a tér minden irányában azonos intenzitással, azaz gömbszerűen, egy irányban párhuzamos sugarakkal azaz hengeresen, és egy irányban széttartó sugarakkal, azaz kúpszerűen. Ezt a **Spherical** (gömbszerű), **Cylindrical** (hengeres) és **Conical** (kúpos) kapcsolókkal tudjuk kiválasztani.

Gömbszerű fényforrásnál nincs probléma, ott minden irányban sugároz a tárgy, de mi a helyzet irányított, hengeres és kúpos fény esetében? Ilyenkor a fény a tengelypontból az Y tengely

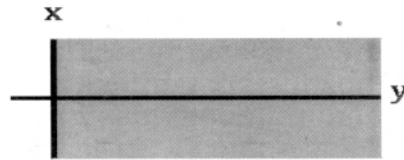
Egyéb attribútumok

Ilyenkor a fény a tengelypontból az Y tengely irányába terjed. A fénykéve átmérőjét vagy a kúp nyílásszögét az X tengely mérete adja meg. Kúpnál ennek az Y-hoz viszonyított mérete számítt, mint az a képen látható.



Kúpos fénykéve

A fényforrás Y tengelyméretének is van jelentősége. Ha a **Diminish Intensity** kapcsolót bekapcsoljuk, a fény terjedése során gyengül. Minél hosszabb az Y tengely, annál hosszabb utat kell megtennie a fénysugárnak, hogy intenzitása felére csökkenjen.



Hengeres fénykéve

Legtöbbször nem használjuk ezt az opciót, de néha például egy gyertya vagy kis lámpás fényének modellezéséhez jól jön. A Diminish Intensity használata növeli a képszámítás idejét.

A fényforrások alapértelmezés szerint nem vetnek árnyékot, csak a **Cast Shadows** kapcsoló bekapcsolása után. Ekkor is csak **Trace** módban készíti el a program a tárgyak árnyékát.

A fényforrásoknak a színét és intenzitását is lehet szabályozni. Egy tárgy színe erősen függ az azt megvilágító fény színétől. Például egy sárga tárgy, amely a piros és zöld színeket veri vissza, zöld és kék fényvel megvilágítva zöldnek látszik, mivel a kéket nem tükrözi, piros fény pedig nem éri.

A fény színét a szokott módon, három színösszetevőjén keresztül lehet megadni. A **Red**, **Green** és

Detail editor

Blue Intensity input mezőben ezek az értékek alapból 255-ön állnak, a fényforrás fehér fényt sugároz. A 255 nem a maximális erősség, ennél erősebb fényforrásokat is létrehozhatunk. Az egyes színösszetevőket **32767**-es intenzitásértékg emelhetjük. Megjegyzem, hogy a 3000 körül erősség már olyan fényt ad, mint egy atomrobbanás.

Az Attributes utolsó opciója a **Genlock**. Ezt bekapcsolva a tárgyat egyáltalán nem árnyékolja és színezi a program, hanem a nullás színnel rajzolja ki. Ha ezután genlockkal egy video jelre keverjük a képet, a tárgy helyén a videojel "átlátszik". Jelenleg csak Amigához létezik genlock, csak azon a gépen lehet a fenti módon képet videojelre keverni. Ennek ellenére a Genlock opció PC-n is működik, a vele készített képet Amigával genlockolhatjuk. Azt, hogy a genlock szín milyen legyen, a Preferences-sel lehet beállítani.

Az Attributes kérdező alján négy nyomógombot találunk. Az OK és a Cancel gondolom nem szorul magyarázatra. A **Load** és a **Save** segítségével a tárgy attribútumait külön is kimenthetjük, betölthetjük. Ez nagy segítséget jelent több azonos attribútumú tárgy elkészítésénél. Létrehozhatunk attribútum könyvtárat a gyakoribb beállításokból, mint króm, üveg, fa, stb, majd ezeket egyszerűen alkalmazhatjuk a tárgyainkra. Mind a Load, mind a Save a szokásos fájl kérdezőt használja. Az Imagine programmal gyárilag is mellékelnek néhány attribútum fájlt, ezek a program könyvtárán belül az Attributes alkönyvtárban helyezkednek el.

Az Attributes fájl maga is egy tárgy, amely mindössze egyetlen Axis-ből áll. Erről meggyőződhetsz, ha egy attribútum beállítsát az **Object/Load**-dal töltesz be.

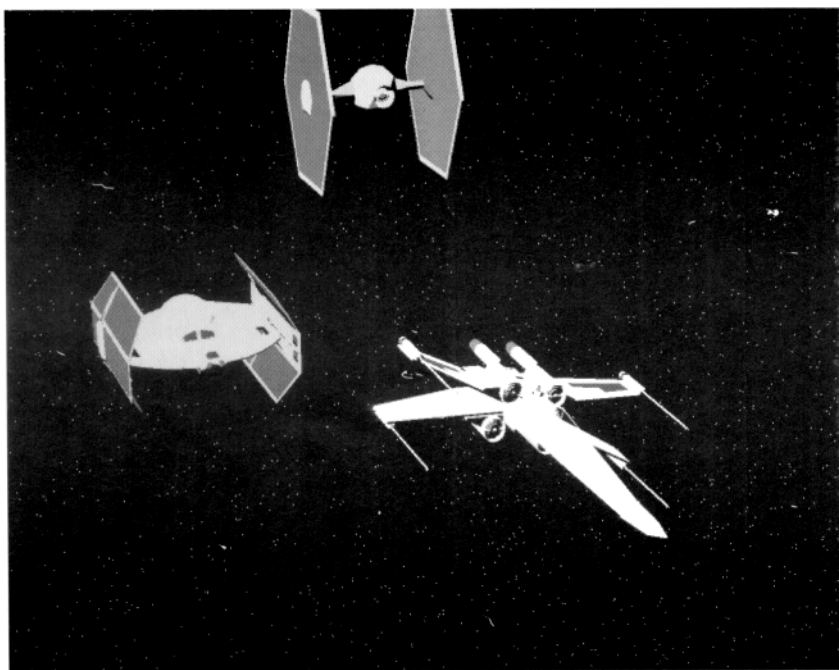
Van ennek valami jelentősége? Igen ez az Imagine egységés koncepciójának a része, lehetőséget ad

Egyéb attribútumok

arra, hogy bármilyen, már létező, kimentett tárgy attribútumait átvegyük. A Load-dal egy object fájlt kiválasztva, annak a tárgynak az attribútumait tölti be a kérdezőbe.

A Load és a Save nem csak a szín, fényvisszaverés, stb. beállításokat kezeli, hanem a tárgyon lévő textúrák és brush-ok paramétereit is, de az önállóan színezett felületelemeket nem.

Az Attributes volt a Detail-lal való ismerkedésünk befejező része. A Functions menüben van három Cycle parancs, amikről nem szoltam, de ezek megértéséhez ismernünk kell a Cycle editor és a ciklikus tárgyakat. A Cycle editor ismertetése után ígérem visszatérünk ezekre.





Az Imagine ray-trace képszámítási módban majdnem teljes fénysugármodellezést csinál. Ha két különböző fénytörési mutatójú tárgyon halad keresztül a fény, akkor megtörik. Ezt a hatást is kezelni tudja a program. A fenti képen a kézinagyító lencséje egy gömb, amelyet egyik tengelye mentén erősen összenyomtam, fénytörési mutatóját pedig 1.52 állítottam, mintha üvegből lenne. Ilyen körülmények között a lencse valóban nagyítóként viselkedik. Ha a tárgy homorú, akkor a hatás fordított, a lencse kicsinyít.

Mivel a fénysugárkövetés fordított irányú, a lencsével nem lehet kondenzálni a fényt még ray-trace-ben sem, mint ahogy a tükörrel sem tudunk fényt vezetni árnyékos helyekre. Ha mégis ilyen képet akarunk készíteni, csalásra van szükség, a hatást nekünk kell modellezni. Erre található példa a 135. oldalon.

Stage editor

Az idáig elkészített tárgyainkat csak a Detail editor Quick Render funkciója által készített képen tudtuk megsejteni. Ennél többre is képes az Imagine, jóval szélesebb lehetőségeket kínál a tárgyak elrendezésének beállítására. A beállítást **stage**-nek, színpadnak szoktuk nevezni. Ebbe nem csak a tárgyak pozíciója számít bele, hanem az animációs mozgások, díszletek, azaz háttérképek, stb.

A stage kezelését a Stage editorban végezzük. Mielőtt ide átlépnénk, kell készíteni néhány tárgyat, amelyekből a színpadot összeállítjuk. A tárgyak elkészítését nem részletezem, csak a végeredményt adom meg. Tehát készíts egy végtelen talajt, fekete-fehér kockás mintával, egy golyót fából és egy kúpot tükörből. Ezek fognak az első igazi képünkön szerepelni.

Ha ezek megvannak, nincs akadálya, hogy a Project menü Stage editor pontjának kiválasztásával belépünk a színpadszerkesztőbe. Ha kívánt szerkesztő helyett egy **"No Project Open"** feliratot kapnál, annak az az oka, hogy nem nyitottál meg tervet. Emlékezz rá, mit írtam a Project editornál, a project tartalomjegyzékében lévő **Staging** fájl tartalmazza a színpad beállításait, enélkül nem tud működni a Stage editor, de a később bemutatandó Action sem. Altvagy megnyitása nem szükséges, arra csak a kép Project editorból való elkészítéséhez van szükség.

A Stage editor hasonló felépítésű, mint a Detail, sok menüpont azonos, de erről már volt szó a könyv elején. Először nézzük át a már ismerős menükbeli különbségeket.

A Project menü egy ponttal bővült. Gondolom mindenki megfejtette már, hogy a **Save Changes** a

Stage editor

változtatások elmentésére szolgál. Ez a funkció írja be a Staging fájlba a színpad aktuális állását. Ha elmulasztjuk használni az editorból való kilépés előtt, a színpadon történt változtatások elvesznek. Erre általában figyelmeztetni szokott a program.

A Display menünek is van egy új pontja, a **Camera View**. Ha ez a pont ki van pipálva, a perspektíva ablakban azt a képet látjuk amit a kamera is, tehát ami a végső képre kerül, függetlenül a Zoom mértékétől.

Hol a kamera? Említettem már, hogy kezdetben a színpadon mindössze egyetlen tárgy van, a kamera. Ezt egy dupla kör jelöli, melyből egy kis vonal áll ki. A vonal jelöli azt az irányt, amerre a kamera néz. A kamera ugyan úgy kezelhető, mint más tárgyak, tengelye van, mozgatható, forgatható, stb. Két dologban azonban különbözik: Nem lehet letörölni és csak egy lehet belőle jelen.

Próbáljunk most valamit összeállítani a színpadon. Ebben a szerkesztőben nem lehet tárgyakat létrehozni, (kivéve az utakat és a fényforrásokat) csak más szerkesztőben elkészített tárgyakkal dolgozhatunk.

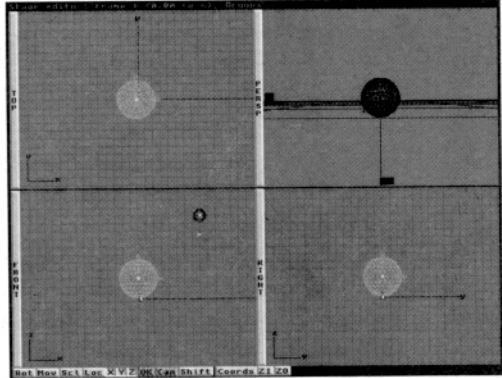
Ennek ellenére a Stage editorban is van egy funkció, amivel a tárgyakat kimenthetjük. Az **Object menü Snapshot** pontja ez. Mi szükség van rá, ha úgyis más editorban már elkészített tárgyakkal dolgozunk? Bizonyos mértékű változásokat tehetünk a tárgyakon a Stage-ben is, effektusok hathatnak a testekre, amelyek deformálhatják, átalakíthatják azokat. Egy ilyen effektus a később ismertetendő robbanás, amely a tárgyat alkotó felületelemeire robbantja szét. A Snapshot a tárgyat az adott képkockában állapotában menti ki, például a szétrobbantott tárgyat széthullott formában. A kimentett tárgy ezután más szerkesztőben tovább alakítható.

Stage editor

Az itt leírtak kissé zavarosnak tűnhetnek, de az Action és Stage editorok megismerése után leegyszerűsödnek.

A **Load** segítségével töltsd be a talajt és a gömböt. A tárgyak nem föltétlen a világ tengelyének origójába fognak kerülni, hanem mindig oda, ahol kimentés-kor voltak. Ez megkönnyíti a relatív pozíciók visszaállítását.

Kapcsold be a **Camera View**-t és rendezd el a két tárgyat. Tapasztalni fogod, milyen nehéz pontosan a kamera látószögébe helyezni a kívánt célobjektet. Ennek megkönnyítésére van egy külön funkció, az **Object/Kamera (Re)Track**.



A funkció előhoz egy kérdezőt, ebbe kell írni annak a tárgynak a nevét, amire a kamerát irányítani szeretnénk. Az enter lenyomását követően a megadott tárgy tengelye kerül a kamera látószögének középpontjába.

A tárgy a Stage editorban ugyan azt a nevet viseli mint a Detail-ban. Ha ezt a nevet elfelejtetted volna, a **Pick/Select** menü **Find Requester**-e segít.

A Pick/Select menü nem tartalmaz új funkciót, sőt, a régiekből is csak kevesebbet, ismertetésük is megtalálható a Detail editornál.

A Camera (Re)Track-nak van egy másik felhasználási módja is, de ezt csak az Action editorral való megismerekedésünk után tudjuk használni.

A Stage editorban a tárgyakat interaktív módon és numerikusan is manipulálhatjuk. Ennek menete

Stage editor

azonos a Detail editorban megismerttel, egyetlen különbség, hogy nem lehet a tárgyak tengelyeit önállóan manipulálni, mert ez műveletet a tárgyszerkesztéshez tartozik. A Transformations-ban nem használható a Transform Axes Only kapcsoló, az interaktív műveletek billentyűi nem veszik figyelembe a shiftet.

Ha kész a Ground és a gömb beállítása, még nincs kész a színpad, fényforrásra is szükség van, hogy lássunk valamit. Fényforrást két módon helyezhetünk el a színpadon: Vagy elkészítjük a Detailban és betöltjük mint normál tárgyat, vagy a Stage editorban hozzuk létre. Előbbi módszert már ismerjük, próbáljuk ki ezért a másodikat.

Válaszd ki az **Object menü Add** pontjának **Light Source** alpontját. Ennek hatására egy olyan kérdező nyílik, mint amit a Detail editor fejezetében az Attributes kérdező Light kapcsolójánál megismertünk.

A paraméterek beállítása után a fényforrás a 0,0,0 pontban jelenik meg, de innen bárhová elmozgathatjuk, mint bármely tárgyat. Az így létrehozott fényforrás a Stage editorban egy hasonló kör jelöli, mint a kamerát. Ha a fény irányított, az irányát egy kis pálcika jelzi, csakúgy mint a kameránál. A hengeres és kúpos fénysugár nyílásszögére és átmérőjére a Detail editornál elmondottak érvényesek, de a tengelyek méretét a Sage editorból nem lehet állítani. Ezt az Action editorban tehetjük meg. A fény most is egy pontból ered. Maga a fényforrás nem látható a renderelt képen, ha erre volna szükség, a Detailban kell létrehozni egy fényforrásként viselkedő tárgyat.

Az Add menüpont két alpontjával hozhatunk létre utakat a Stage editoron belül. Ezek az utak teljesen azonosak a Detailban készítettékekkel, létrehozásuk és szerkesztésük szinte azonos.

Lássuk egy példát a Path létrehozására és szerkesztésére. Hozz létre az **Open Path**-szal egy nyitott utat. A menüpont aktiválása után egy kérdező figyelmeztet arra, hogy fájlnévet kell megadni az úthoz. Az **OK**-ra klikkelés után megjelenik a szokásos fájl kérdező, ahol a névmegadást végezhetjük. Ha ez megvan, megjelenik a szerkesztőben az út. Válaszd ki ezt (a Stage editorban nem lehet Drag Box-ot vagy Lasso-t használni a kiválasztáshoz, csak a Clik-et) majd a **Mode menü Edit Path** pontjának segítségével lépj át útszerkesztő módba. Ebben az előzőleg kiválasztott utat szerkeszthetjük a most aktiválódó Path menü parancsainak a segítségével.

Az út kontroll pontjait a move és rotate interaktív manipulációkkal vagy a Transformation kérdezőn keresztül tudjuk befolyásolni, a már megismert módon. Útszakasz elfelezését, új kontroll pont létrehozását a **Path menü Split Segment**-jével érhetjük el, míg pontot törölni ugyanezen menü **Delete Point**-jával lehet. A **Save Path**-szal menthetjük ki a szerkesztett utat Imagine object-ként. A fájl nevét megváltoztathatjuk a megjelenő kérdezőben.

Ha az útszerkesztést befejeztük a Mode menün keresztül vissza kell lépni a Stage editor másik módjába a **Pick Groups**-ba. Amikor ezt tesszük, megjelenik egy kérdező, amelyet arról érdeklődik, hogy ki akarjuk-e menteni az imént szerkesztett utat. Ha igen, a szokásos fájl kérdező jelenik meg. Ha nem mentjük ki a Path-t, az akkor is megmarad a szerkesztőben, de ha szerkesztés közben sem volt kimentve, a változtatások elvesznek, az eredetileg megadott néven csak az alapértelmezés szerinti út tárolódik.

Most még nem tudjuk mire használni a Path-t, csak az animációs mozgáspályák létrehozásánál lesz

Stage editor

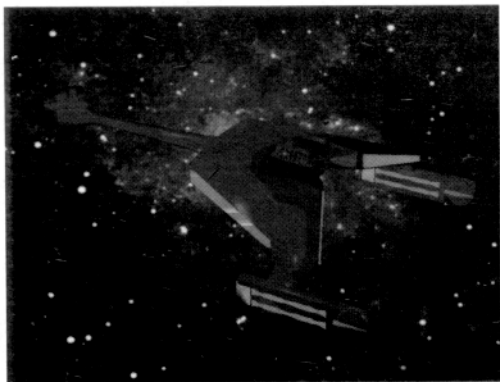
szerepük, de erre várni kell az Action editorral való megismerkedésünkig.

Az utakkal kapcsolatos az **Objects menü Show Path Length** menüpontja, amely egy kérdező ablakban megjeleníti a kiválasztott út aktuális hosszát Imagine egységben.

Ha már az Add alpontjainál járunk, ismerkedjünk meg az **Add Axis**-szal is. Ez a szokásos tengelyt hozza létre. Mi szükség erre, ha a Stage editorban nem is lehet tárgyakat létehozni? Ezt az Axis-t nem lehet kimenteni, (csak a Snapshot-tal), ennek ellenére tárgyként viselkedik, adatait maga a Staging fájl tárolja. Mivel a renderelt képen nem látszik, kitűnően használható a kamera nézőpontjának meghatározásához. Ez a tengely alpból a Track nevet viseli, erre irányítva a kamerát, az bárhová tekinthet. Animáció közben mozgatva a kamera nézőpontja is mozogni fog, de erről részletesen később.

Ahhoz, hogy a Stage editorban egy állókép beállításait megtegyük, elegendő ennyit tudni. Javaslom próbálgasd a lehetőségeket, készíts néhány képet.

Bonyolult tárgy gyorsabb megjelenítéséhez most is használható a **Quickdraw** az **Object** menüből.



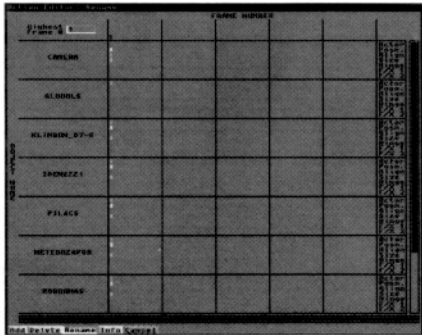
Minden betöltött tárgy a Detail editorban, az Attributesben meghatározott módon rajzolódik ki, normál módon, vagy befoglaló kerettel, de ezt tetszés szerint megváltoztathatjuk az említett menüponttal.

Action editor

A ray-tracing csúcsa az animáció készítés. Az Imagine ehhez is nagyon sok segítséget nyújt, olyan funkciói vannak, melyekkel könnyedén készíthetünk látványos animációkat.

Az Animációk elemeinek összeállítására szolgál az Action editor, ahol állóképekkel kapcsolaban is találunk néhány hasznos lehetőséget. Ebbe a szerkesztőbe is csak úgy léphetünk be, ha Project van megnyitva.

Belépés után egy új képernyő fogad bennünket. Ez nem a szokott négy nézetes elrendezés, itt minden beállítást kérdészkön keresztül numerikusan tehetünk meg. A kép bal felső sarkában a **Highest Frame#** input mezőben adhatjuk meg az animáció képkocká-



inak számát. Amígán max. 9999 képkockából állhat az animáció, míg PC-n a PC-DOS korlátai miatt csak 999-ből. Vele egyvonalban, a **Frame Number** felirat alatt lévő számsor a képkockákat jelképezi. Itt mindig csak annyi szám jelenik meg, amennyit a Highest Frame#-nél beállítottunk. Ugyannyi kockát jelképező szám fog feltűnni a Project editorban is a Stills-nél. A könnyebb eligazodás érdekében tízessével oszlopokra vannak bontva a képkockák.

Egyszerre ötven látható, hosszabb animáció esetén a képernyő alján, az User Gadgetek felett lévő gördítő léccel lehet lapozni.

A Highest Frame# alatti oszlopban a Stage-ben lévő tárgyak nevei találhatóak. Kezdetben, új színpadnál

Action editor

kettő, a mindig jelen lévő **Camera** és a **Globals**. Az utolsó sorban lévő (**new**) nem object, hanem ide tölthető be új tárgy.

Minden tárgy hétféle paraméterrel rendelkezhet az Action-ban, ezeket jelölik az **Actor**, **Posn.**, **Size**, **Hinge**, **F/X1** és **F/X2** sorok. Ha valamelyik paraméter be van állítva, azt a sorban, a megfelelő kockák alatti színes vonalak jelölik. A színeknek jelentésük van, ezeket később ismertetem.

A képernyő alján öt user gadget található. Az **Add**-ot bekapcsolva új paramétert helyezhetünk el, a **Delet**-tel pedig törölhetünk. Nem csak egyes paramétereket állíthatunk be és törölhetünk le, hanem egész tárgyakat is.

A **Rename** bekapcsolása után a tárgyak nevét, ami az első oszlopban található, változtathatjuk meg. Ez úgy zajlik, hogy a névre klikkelünk, majd a megjelenő kérdezőbe beírjuk az új nevet és lenyomjuk az entert. A továbbiakban a tárgy az Action és Stage editorban ezen a néven szerepel, de a tárgy valódi neve változik meg.

A Camera és a Globals neveket ne változtatsd meg mert ez bonyodalmakhoz vezet!

Az **Info**-t bekapcsolva a már beállított paraméterek ablakait kapjuk vissza, megtekinthetjük, de akár meg is változtahtjuk az itt szereplő értékeket.

A **Cancel** gomb csak az **Add** használatakor aktiválódik, segítségével a már megkezdett Add műveletet vonhatjuk vissza.

Az Action editornak két menüje van, az első a Project csak annyiban különbözik a Stage editor azonos menüjétől, hogy az Action editor helyett a Stage editor neve áll. A másik menü a **Functions** nagyrészt a már ismertetett User Gadget-ekkel azonos nevű és funkciójú menüpontokból áll.

A **Find** akkor hasznos, ha annyi tárgy szerepel a

színpadon, hogy egyszerre el se férnek a képernyőn. Ekkor a jobb oldali gördítőléc segítségével lapozhatunk közöttük, de a Find kérdezőjébe beírva a keresett tárgy nevét, úgy áll be a gördítőléc, hogy a megadott object a képernyőre kerüljön.

A **Short** használatával ABC rendbe rendezhetjük a tárgyakat, áttekinthetővé téve a listát. A rendezés nem vonatkozik a Camera és a Globals tárgyra azok mindig a lista elején maradnak.

A **Cancel Add** menüpont a Cancel gadgettel azonos.

Miután megismerkedtünk az Action editor felépítésével, belemerülhetünk a részletekbe is.

Az első paramétercsoport az **Actor**. Ez a tárgyat "alakító" színészt, vagyis a hozzá tartozó fájlt jelenti. A Camera-nak nincs Actor paramétere, mivel az állandóan jelen van.

A Globals-nak már van Actor jele, bár ez nem a szokásos object, hanem a színpad környezetének leírása, a díszlet. Mielőtt megnéznénk, terjeszd ki az animáció hosszát mondjuk 20 képkockára. Ez igen egyszerű dolog, mindössze a Highest Frame# mezőbe kell beírni a számot. Ha az animációt rövidítjük, a fölöslegessé vált kockák minden beállítással együtt elvesznek!

Info módban klikkelj a Globals Actor jelére. Egy tekintélyes méretű kérdező tűnik fel, **Globals Info** néven. Tetején a **Start Frame** és az **End Frame** után két input mezőt találunk. Ezek azonosak az Action editor minden kérdezőjében, azt a részt jelentik az animációban, ahol a kérdező beállításai érvényesek.

Most itt mindkét érték 1, vagyis a beállítások csak az 1. képkockára vonatkoznak. Bővítsük ezt ki mind a 20 kockára, azaz End Frame-nek írd be 20-at. Ne feledd, hogy minden megváltoztatott paramétert az enterrel kell véglegesíteni!

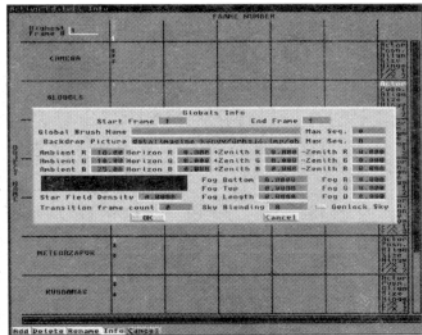
Action editor

Kissé lentebb a **Global Brush Name** input mezőben környezetkép megadására nyílik lehetőség. Ezt a képet a program a kamera mögé teszi fel. A kamera hátrafelé nem lát, akkor mi értelme a környezet képének? Az, hogy a tükröződő tárgyakban látszik, ennek a képnek az alkalmazásával növelni lehet az életszerűséget. A "Csendélet" című kép a 168. oldalon demonstrálja ezt. A kép Global Brush-a egy bedigizett fotó volt, amely engem ábrázolt fényképezés közben. Ez a kép tükröződik a tárgyakban, olyan hatást keltve, mintha az egész kép egy fotó lenne.

A **Backdrop Picture** hasonló feladatot tölt be, de ez a kép mindig a kamera látóterébe, a tárgyak mögé kerül, mint egy színpadi díszlet. Segítségével hozhatunk létre tájkép, felhős ég, stb háttérrel. A háttér kép használatára van egy fontos megkötés: méretének, felbontásának és színei számának meg kell egyeznie a renderelendő kép azonos értékeivel, ellenkező esetben a brush nem kerül rá a képre.

A Global Brush és a Backdrop Picture tükröződéseit Scanline algoritmussal is elkészíti a program.

Midkét képet animálhatjuk is például, a Backdrop Picture egy eget ábrázolhat, amin átúsznak a felhők. Ennek a megvalósításához szükségünk van egy háttér animációra, amit nem anim formátumban, hanem képenként kell kimenteni. Minden képnek azonos nevet kell viselni, ugyanabba a könyvtárban kell lenni, kiterjesztésnek pedig .0001, .0002, .0003,....stb kell adni. PC-n a DOS megkötése miatt csak három karakterből állhat a kiterjesztés, de az

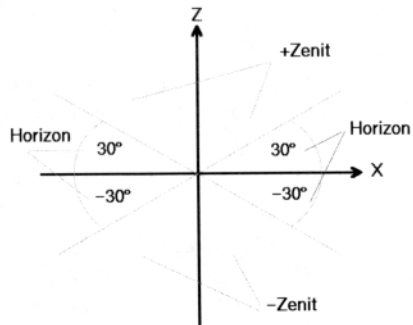


elv ugyan az. A Global Brush és a Backdrop Picture input mezőben csak a kép nevét kell megadni, kiterjesztés nélkül és pont nélkül, a képkockák maximális számát pedig a **Max Seq.** mezőben kell beírni. Az animáció képeinek számítása során minden képhez a következő Global vagy/és Backdrop kép használódik fel. Ha az anim hosszabb, mint a Max Seq-ban megadott hossz, a képek elfogytával újból az első következnek, akár csak a Detail editorban tárgyra feszített animáló brush-nél.

A valóságban nem csak a direkt fények világítják meg a tárgyakat, hanem a legtöbb esetben a különböző felületekről, falakról, stb visszaverődő szórt, ún. diffúz fények is. Az Imagine képes ennek a fénynek a modellezésére. A diffúz fény erősségét és színét az **Ambient R,G,B** értékeivel szabályozhatjuk. Erősebb diffúz fényben a tárgyak világosak lesznek, csökken a megvilágított és az árnyékos részek közötti fénykülönbség, de csökkennek, sőt el is tűnhetnek a vetett árnyékok is. Hogy milyen legyen a szórt fény erőssége, nehéz megmondani, a környezetről, a kép témájától is függ, de gyakorlatilag a színösszetevőnként 40-50-es értékek már magasnak számítanak.

Képeink háttérét nem csak a Global Brush alkalmazásával tudjuk létrehozni, hanem színátmenetet is definiálhatunk a helyére. Ez csak akkor működik, ha nincs Global Brush, mert az erősebb, elnyomja az égszínt.

A háttér három rétegre



Action editor

van osztva, ezek mindegyikének saját színe van. A **Horizont R,G,B** a horizontvonal és a fölötte-alatta lévő 30°-30°-os rész színéért felelős. A **+Zenit R,G,B** a Horizon feletti rész színe, a **-Zenit R,G,B** pedig a Horizon alattié. A színek beállításában segít a kérdező bal-közép táján lévő nagy téglalap.

A természetben ezek a színrétegek nem különülnek el élesen egymástól. Az Imagine is össze tudja mosni az egyes színrétegeket. Az összemosás mértékéért felelős a kérdező alján lévő **Sky Blending** input mező és annak tartalma. Ha ez 0, nincs összemosás, míg a 255 a legjobb színátmenetet eredményezi.

A másik háttérképzéssel kapcsolatos lehetőség a csillagos ég szimulálása. Sajnos ez az opció nem tökéletes, a csillagok véletlenszerűen kerülnek fel az égre (a Horizon és +Zenit-re egyaránt) minden képkocka számításakor, ezért animációban nem lehet használni, illetve nem ad szép eredményt. A másik hiányosság, hogy a csillagok átlátszó tárgyakon keresztül, vagy tükröződő tárgyokban nem láthatók.

A csillagok színét a Preferencesben lehet beállítani, míg sűrűségüket a **Global Info** kérdező **Star Field Density** input mezőjében. A sűrűség mértéke legfeljebb 1 lehet, de már a néhány ezredes érték is elegendő csillagot rak az égre.

A csillaggenerálást kikapcsolni a Star Field Density 0 értékével lehet.

A háttér képzésével összefüggő utolsó opció a **Genlock Sky**. A kapcsolót kiikszelve sem a Global Brush-t, sem a színátmenetet sem a csillagokat nem készíti el számításakor a program, helyette mindezeket a Genlockolható nullás színregiszter színével rajzolja ki. Ezt a képet Amigán genlockkal video jelre keverve, a háttér helyén a videojel látszik.

Az Imagine-nel globális ködöt is hozhatunk létre.

Ez a kód a **Fog Bottom** mint alsó és **Fog Top** mint felső magassághatárok között jön létre. Mindkét értéket Imagine egységben kell megadni. **A Fog Lenght** szintén Imagine egységben megadandó érték, a kód sűrűségére van hatással. Ekkora távolságon gyengül a kődön áthaladó fény eredeti intenzitásának két harmadára.

A kód színét is befolyásolhatjuk a **Fog R,G,B** input mezőkön keresztül.

A Global Info utolsó input mezője szintén sok kérdezőjében megtalálható az Action editornak, az Imagine egy nagyszerű képességével, a metamorfózis készítővel áll kapcsolatban.

Mi az a metamorfózis? Gondolom mindenki tudja, hogy ez a szó átalakulást, átváltozást jelent. Pontosan ezt jelenti az Imagine-ben is. A program képes egy tárgyat egy másikba alakítani. Az átalakítás a tárgy minden tulajdonságán végbemegy, tehát az alakon, színeken, textúrákon, felületi tulajdonságokon, stb.

A morfózis készítésére van néhány megkötés: a tárgy alakjának átalakításakor a kiinduló és cél objectnek azonos számú pontokból, élekből és felületekből kell állni. Elméletileg ennyi is elég, gyakorlatilag azonban több szükséges az igazán szép átalakításhoz. Az alak átváltoztatása közben a kiindulási tárgy minden pontját a céltárgy azonos sorszámú pontjának pozíciójába viszi át a program, a pontok sorrendjének tehát azonosnak kell lenni, ellenkező esetben kissé kusza lesz az átalakulás. Ennek a feltételnek legkönnyebben úgy tudunk megfelelni, ha a céltárgyat eleve a kiindulási tárgyból hozzuk létre annak torzításával, pontjainak áthelyezésével. Ez a probléma nem merül fel, ha csak a tárgy attribútumait változtatjuk, például egy pöttyös gömböt átalakítunk egy fa golyóvá.

Action editor

Első próbaként készítsünk egy napfelkeltét. Ehhez nem lesz szükségünk egyetlen tárgyra sem, a Horizon és a +Zenit színeit fogjuk változtatni. Írd át a Start és End Famet-t úgy, hogy ez a beállítás csak az első képkockára vonatkozzon. A Horizon és a +Zenit színe legyen egyaránt fekete. Klikkelj az OK-ra, amivel kilépünk a kérdezőből.

Ezután kapcsolj be az Add-ot, majd vidd a kurzort a Gobals Actor sorában a 2. képkocka alá. A pontos beállítást segíti, hogy mind a jobb oldali oszlopban a paramétertípusoknál, mind a képkockák számának sorában invertálódik az a paraméter típus és képszám, amelyek kereszteződésében a kurzor áll. Klikkelj ide egyet. Az állapot sorban megjelenik egy felirat, miszerint a kezdő frame a 2. és válaszd ki a befejező frame-t. Menjen végbe a naplemente a teljes animáció alatt, vagyis az End Frame legyen 20. újból megnyílik egy, az előzőhöz hasonló kérdező (ez nem ugyanaz!) ahol a horizontnak adj egy pirosas rózsaszín árnyalatot, a +Zenitnek ugyan ezt csak sötétebben. Ezt a szint fogjuk elérni az utolsó képkockában, a közte lévőkön pedig az előző kérdezőben beállított fekete és ezek közti átmenetet hozunk létre.

Azt, hogy az előző és az itteni értékek között az átalakulás hány lépésben menjen végbe, a **Transition Frame Count** határozza meg. Ha ez nulla, az átalakulás azonnal végbemegy, vagyis ebben az esetben a második képkockán már a rózsaszínes eget látjuk.

Nyújtsuk el ezt az átalakulást, amilyen hosszúra lehet. Mivel az első kockán az előző kérdezőben meghatározott értékek vannak, a huszadikban pedig már az itt megadott paraméterekkel kell rendelkeznie a Globals-nak, marad 18 képkocka az átalakulásra, ezt írd be a Transition Frame Count-hoz.

Persze lehet kevesebbet is írni, akkor hamarabb felveszi a tárgy, jelen esetben a Globals, a végső értékeket. Ha kész, klikkelj az OK-ra, a beállítások érvényessé válnak. A Globals Actor sorában az első kocka alatt lesz egy narancssárga jel, amely a paraméterek beállítottságát jelképezi. A másodiktól a huszadik kockáig terjed egy vonal, amely a 19. kockáig világosabb. Ez jelzi, hogy ezen a szakaszon a paraméterek átalakuláson mennek keresztül, majd a huszadik kocka alatti narancs színnél megállapodnak. Ha a Transitionhoz alacsonyabb értéket írsz, a világosabb sáv rövidebb lesz, pontosan jelölve az átalakulás hosszát.

Ezeket a színes jeleket, vonalakat **Time Line Display**-nak, időfolyam megjelenítőnek, vagy felsoroló jelnek nevezzük, segítségükkel jól átlátható az animáció. Lépj vissza Info módba és kattints bele a **Camera Posn.** sorában az első kocka alatt lévő zöld jelre. Ez a paramétertípus a tárgyak pozíciójáért felelős. A megjelenő kérdezőben a Start és End Frame input mezők jelentése a szokásos. Az **End Frame Position Values** melletti három input mezőben a tárgynak a világ koordináta rendszerében elfoglalt pozícióját adhatjuk meg.

Ha nem ott lenne, állítsd a kamerát az origóba, azaz a 0,0,0 pozícióba, majd klikkelj az OK-ra. Az, hogy a pozíció jel csak az első képkocka alatt van nem baj, ez a kamera, vagy más tárgy megjelenésére nincs hatással, az csak az Actor sor jeleitől függ. A kamera pozícióját az első kockákban beállítjuk, utána ezen már nem változtatunk, ezért nem szerepel jel a 2-19. kockák alatt.

Ez általánosságban is elmondható, felsoroló jel (kivéve az Actor-t) csak ott szerepel, ahol paramétert változtatunk. Még mindig info módban kattints a kamera **Align** sorában lévő felsoroló jelre. Ez a

Action editor

tárgyak irányáért felelős, a kameránál a nézői-ránnyal azonos. A kérdező alig különbözik a pozíció kérdezőjétől, itt az **End Frame Alignment Value** melletti mezőkben a tárgy és a világ azonos tengelyeinek egymással bezárt szögeit lehet megadni, csakúgy, mint a Transformation kérdezőjében. A kamera az Y tengely mentén pozitív irányba tekint. Legyen mindhárom érték nulla fok.

Egy igazi kameránál a látószög fontos dolog, annak változtatásával rá lehet közelíteni tárgyra, vagy el lehet távolodni azokról. A látószöget az optika fókusz távolságának változtatásával lehet befolyásolni. Minél hosszabb a fókusztávolság, annál kisebb a látószög, tehát a kamera annál messzebb "lát". Ez nem ugyan az az eredmény, mintha közelebb vittük volna a kamerát a tárgyhoz, mivel akkor torzulhat a perspektíva látványa.

A fókuszt egészen lecsökkentve ún. halszemhatás következik be, vagyis a kamera látószöge közelít a 180° -hoz. Ez már jelentős torzulást okoz a képen, de sok esetben épp ez a lényeg.

Az Imagine kamerája pontosan így működik, a fókusztávolságának állításával változtatni tudjuk a látószöget.

A fókusztávolság a kamera X és Y tengelyeinek aránya, ugyan úgy, mint a kúp fényforrásnál láttuk. Az Y hosszát növelve a kamera látószöge szűkül, míg csökkentve szélesedik.

Az Action editorban a tárgy méretéért a **Size** sor felelős. Mivel a kamera egyetlen tengelyből áll, így ez arra vonatkozik, normál objektumnál viszont a valós méretre.

A kamera alap helyzetben 1:2 fókusztávolságra van állítva. A Z méretének nincs jelentősége. Állítsd a fókusztávolságot 1:1-re, majd klikkelj az OK-ra.

Ide kívánczodik egy fontos dolog. Bár a Globals-nak

nincs Size jele, mégis van egy alap mérete, ami mindhárom tengelyen 1024 egység. Ennek csak trace képszámítási módban van jelentősége, amikor a program a gyorsulás érdekében csak egy, világ origója középpontú, tengelyenként ekkora sugarú ellipszoidon belüli területet vesz figyelembe a tárgyak letapogatásánál. A lámpák, a kamera és a célpontok minden következmény nélkül kívül eshetnek ezen a területen, de a tárgyak kilógó részei hiányozni fognak a képről. Ha nagyobb színpadot használsz, új felsorolójel hozzáadása után állítsd be a szükséges méretet.

A képek elkészítéséhez mindenképpen szükségünk van fényforrásra, ami lehet fényforrásként viselkedő tárgy, vagy Light Source. Az előbbi a normál tárgyakkal teljesen azonos módon kezelhető, fényének paramétereit a Detail editorban az Attributes kérdésből tudjuk befolyásolni.

A **Light Source** egy speciális tárgy, a Stage vagy az Action editorban lehet létrehozni, és a Staging fájlban tárolódik, csakúgy mint az előbb megismert Axis. Paramétereinek állítására az Action editorban nyílik mód.

A Stage editorban már láttuk hogyan lehet Light Source-t készíteni, az Action-ban ez másként megy. Keresd meg azt a sort, amelyben a **(new)** felirat áll. Váltás át a Function menüben, vagy a gadgettrel az Add-ra. Ahhoz hogy új tárgyat tegyünk a színpadra, a hozzá való színészt, azaz Actor-t kell beállítani. Vidd a kurzort az Actor sorban az első képkockát jelképező szám alá, majd klikkelj egyet a bal gombbal. Az állapot sorban megjelenik a "**Starting frame is 1. Choose ending frame**" felirat, ami jelzi, hogy a kezdő kocka az 1-es, válaszd ki a befejezőt. Vidd a kurzort ugyan ebben a sorban a huszadik kockához és ismét klikkelj egyet. Megjelenik a

Action editor

Specify Type kérdező, három választási lehetőséget kínálva. A **Normal Object** segítségével lemezre mentett tárgyat tölthetünk be a színpadra, az **Axis**-szal pedig tengelyt hozhatunk létre. Nekünk most a középső gombra kell klikkelni, amelyikben a **Light Source** felirat van.

A (new) helyén megjelenik a Light Source név, néhány felsorolójel, és a Detail editorban az Attributes Light opciójánál ismertetett kérdező, amely itt három plusz input mezőt tartalmaz. A Start és End Fame jelentését már ismerjük, ha véletlenül rosszul jelöltük ki a színész jelenésének idejét, itt módosítani lehet azt.

A **Transition frame count** a morfózissal van kapcsolatban, mint azt a Globals-nál magyaráztam. Megtehetjük, hogy egy lámpa fénye, sugárzásának típusa, iránya, stb folyamatosan változzon át. Az elv a Globals-nál ismertetettel azonos. Hosszabb animációknál az Actor felsorolójelet többször megszakítva a fényforrásunk (de bármelyik tárgy is) állandóan változtathatja paramétereit, mint egy diszkólámpa.

Jelen animációnkban a lámpa végig azonos, fehér fényt sugározzon, hagyj tehát a beállításokat alapon, úgy klikkelj az OK-ra.

Ha később változtatni akarunk az értékeken, bármikor megtehetjük az Info bekapcsolása után.

A lámpánk pozícióján változtassunk, emeljük fel jó magasra. Info-ban lépj bele a lámpa **Posn.** felsoroló-jelébe és a Z értékét ird át 1000-re. Mivel a lámpa gömbszerűen sugároz, az Alignment-nek nincs jelentősége.

Ezzel minden kész az animációs beállításokból, mentsd ki a változtatásokat és lépj vissza a Project editorba.

Animáció összefűzése

Ha eddig nem tetted volna meg, altervet kell nyitni a **New** gombbal. Ha van alterv nyitva, a **Range** segítségével jelöld ki az összes képkockát, majd a **Generate**-tel indítsd el a képszámítást.

A géped sebességétől függően néhány perc vagy néhány tíz perc számolás után elkészülnek a képek. Ha menet közben meggondolnád magad, a számoláskor megjelenő **Cancel** gombra kattintva megállíthatod a folyamatot. Ez nem mindig marad abba azonnal, a gép még befejezi az aktuális műveletét.

A **Cancel** után egy kérdező jelenik meg, miszerint törölje-e a félbemaradt képkockát. Ha a **No**-ra klillelsz, rövid "takarítás" után a félkész képet lezárja a program, a neki megfelelő szám alatt megjelenik a csillag, a továbbiakban úgy kezeli a, mintha teljesen kiszámolt kép lenne.

Ezek még csak önálló képek, az animációt külön kell összefűzni. Jelöld ki az összes képkockát, majd kattints a legalsó sor gombjai közül a **Make** feliratra. Ügyelj arra, hogy a **Generate New Cells Only** kapcsoló be legyen kapcsolva, ellenkező esetben minden kockát újból kiszámol a program. Ha a **Make**-et úgy használjuk, hogy a kijelölt kockák között kiszámolatlan is van, azt automatikusan elkészíti a program, mikor az illető képre kerül a sor.

Az animáció elkészítése attól függően megy végbe, hogy milyen anim formátumot használunk. Tegyük fel, hogy **Imagine**-t. (Ez a **Modify**-jal változtatható meg, ha **Anim**-et, vagy **Flick**-et ikszeltél volna az alterv nyitásakor.) Ebben a formátumban lehetőség van animációs vezérlő fájl készítésére, amivel szabályozhatjuk az egyes képkockák animációba kerülésének sorrendjét. Ez majd egy későbbi téma

Animáció összefűzése

lesz, most olyan sorrendben használjuk fel a képet, ahogy elkészítettük.

A Make használata után megjelenik egy kérdező ami megkérdez, mivel nem talált **Movie** (ez a vezérlő-fájl) fájlt, készítsen-e egy átmenetit a kijelölt kockákból. Ha a No-ra kattintunk, nem készíti el az Animációt, ha a Yes-re, akkor a kockákat eredeti sorrendjükben használja fel.

A következő kérdés, a "**Make a looping movie?**" arról érdeklődik, hogy loopolható, folyamatosan játszható animációt akarunk-e.

Ezután újabb kérdező tűnik fel, a "**Lock the palette?**" Itt eldönthetjük, hogy az animáció képkockáihoz felhasznált paletta végig azonos legyen-e.

Bizonyos animáció-lejátszók nem képesek arra, hogy az anim képkockái között a palettát is kicseréljék, ezekhez hasznos ez a lehetőség. Ha a képek már el vannak készítve, a Lock palette nem működik, mivel akkor a képek palettája már adott. Használatakor vagy ki kell kapcsolni a Generate New Cells Only kapcsolót, vagy még kiszámolatlan kockákból kell felépíteni az animációt. Mindkét esetben az animáció készítése közben számolódnak ki a képkockák, a Lock Palette hatása alatt.

Ez az elmélet, gyakorlatilag az általam tesztelt változatban nem működött. Sajnos.

Mielőtt az animációkészítés tényegesen elindul, még egy kérdezőben válaszolni kell a "**Delete picture files after use them?**", vagyis töröljem a képeket felhasználásuk után? kérdésre. Ha itt Yes-szel válaszolsz, a képeket animációba kerülésük után törölni foja a program. Ez hasznos, ha kevés a hely a winchesteren, vagy Amigán floppyról használva a progamot. Ha van elegendő háttérkapacitás, válaszd inkább a No-t, még ha nem is kellene a képek csak az anim, ugyanis ha közbejön egy áram-

Animáció összefűzése

szünet, vagy a gép lefagy, az animációkészítést újra kell kezdeni, amihez szükség van a korábbi, egyszer már felhasznált képekre is.

Mostmár tényleg elindul az animáció készítés, a program sorban megjeleníti a képeket majd beviszi azokat az anim fájlba. A bevitel előtti megjelenítésnek az az előnye, hogy így olyan képekből is tudunk lejátszható animációt készíteni amely képek megjelenítésére egyébként nem is alkalmas a video rendszerünk. Például 24 bites képekből készíthetünk animációt, még ha nincs is 24 bites kártya a gépben. A képeket megjelenítéskor a gépen elérhető legjobb képernyőtípusra konvertálja át a program, majd úgy viszi be az animációba. Amígán ez legtöbb esetben a 4096 színű HAM, PC-n a 256 színű VGA.

Az Imagine formátumú animáció a megnyitott alterv könyvtárán belüli anim könyvtárban helyezkedik el képkockánként külön delta fájlban anim.0001, anim.0002, anim.0003, stb néven. PC-n a kiterjesztés csak három számjegy.

Ugyan ebben a könyvtárban találunk egy script nevű fájlt, ami az animáció-lejátszó számára tartalmaz információkat.

A Movie vezérlőfájlt az alterv könyvtárában találjuk. Ez egy normál szöveg fájl, amely kulcsszavakból tevődik össze. A fájl csak az összefűzés sorrendjére van hatással, a lejátszásra már nem. Szerkezetét nemsokára ismertetem.

Az elkészült animációinkat kétféle módon tekinthetjük meg. Az egyik hogy a Project editorban a Movie gombjai közül a **Load**-dal betöltjük, majd az aktiválódó **Play Once**, vagy a **Play Loop**-ra klikkelve egyszer, vagy ismétlődve megnézhetjük. Az animáció lejátszásának sebességét a funkció gombokkal szabályozhatjuk. Az F1 60 frame/s az F2 30, F3 15, F4 10 frame/s sebességet jelent, a többi még alacsonyabb.

Animáció összefűzése

nyabbat. Az F10 leütése után másodpercenként mindössze egy kockát mutat meg a lejátzó. A space segítségével manuális módba válthatunk, a space minden megnyomására egy kockányit halad az anim. A normál módra visszatérni a funkció billentyűkkel lehet.

Az animációból való kilépésre az Esc. billentyű szolgál.

Ha az animációt loopolhatóan fűztük össze, a Play Once is folyamatosan játsza azt.

Az animáció megtekintésének másik módja az Imagine-hez mellékelt PlayIANM animáció lejátzó program. Ezt a DOS-ból használhatjuk, paramétereinek a lejátyszandó animációt tartalmazó anim könyvtárat kell megadni, például:

PlayIANM Data:Falcon.imp/Prev.pix/anim

A lejátzás sebességét ugyan úgy szabályozhatjuk, mint az Imagine-ben.

Az animációt a Drop gombbal tudjuk törölni a memóriából, felszabadítva az általa lefoglalt helyet.

Animációs vezérlő fájl

Ha már ennyi szó esett a vezérlő fájlról, lássuk hogyan is néz az ki.

Klikkelj az **Edit** gombra, mire megjelenik a "**No movie file-make temporary file from pick list?**" kérdező. Ha voltak kijelölt frame-ek, azokból most átmeneti Movie fájl készült, mint a Make esetében, de ha nem volt kijelölt képkocka, akkor erre figyelmeztet.

Klikkelj a Yes-re, majd válaszolj a már ismerős kérdezőben. Ezután egy szövegszerkesztő töltődik be,

Animációs vezérlőfájl

amelyben ott lesz az átmeneti Movie fájl. Ebben a fájlban néhány kulcsszó segítségével adjuk meg a képek összefűzési sorrendjét.

A következő kulcsszavakat használhatjuk:

Scene név - Animációs alegységet határoz meg. Később a "név"-re hivatkozva az alegység hajtódik végre.

Play - Egy vagy több képkocka vagy alegység lejátszását végzi. Például:

Play - lejátsza az összes képkockát, azok eredeti sorrendjében.

Play X - lejátsza az X sorszámú képkockát

Play X-Y - lejátsza az X-től Y-ig terjedő kockákat. Ha X a nagyobb, akkor visszafelé halad.

Play név - a "név" nevű alegységet játsza le

A Play-hez kapcsolódik két kulcsszó. Az **X Times** X-szer ismétli az adott Play-t, tehát a

Play 1-10 2 Times

kétszer egymás után megjeleníti a képeket 1-től 10-ig.

A **Forever** végtelen ciklusban ismétli az adott Play-t, a **Make Looping Anim**-re adott **Yes**-szel egyenértékű.

Movie - Az animációs rész kezdete. Mindig ez előtt kel a Scene-nel létrehozni az alegységeket, amiket ténylegesen csak a Movie után jelenítünk meg Play név-vel.

Lock - A Lock Palette kulcsszava.

Mindezek ismeretében hozz létre a 20 kockából egy olyan animációt, amely az első tíz kockát kétszer

Animációk összefűzése

lejátsza oda-vissza, majd lejátsza a tizenegyedikről a huszadikig és az egészet ismétli végtelenítve.

Ellenőrzésképpen ez valahogy így néz ki:

Scene valami

Play 1-10

Play 10-1

Scene ezaz

Play valami 2 Times

Play 11-20

Movie

Lock

Play ezaz Forever

A fájl kimentése a használt szöveg szerkesztőtől függ, erre most nem térek ki. Ha nem felelne meg a program által felkínált text editor, később ismertetem a módját, hogy másikat használj a programból, addig viszont szerkeszd a Movie fájlt egy neked tetsző szövegszerkesztővel, a programon kívül.

Mivel a Movie fájl csak az összefűzés sorrendjére van hatással, a többször felhasznált képek többször is bekerülnek az animációba, növelve annak méretét.

Ha az alterv könyvtárában van anim vezérlő fájl, a Make csak a "Delete picture affer use them?" kérdést teszi fel, a többi beállítást a Movie fájlból olvassa az Edit pedig kérdésés nélkül a már létező fájlt nyitja meg szerkesztésre.

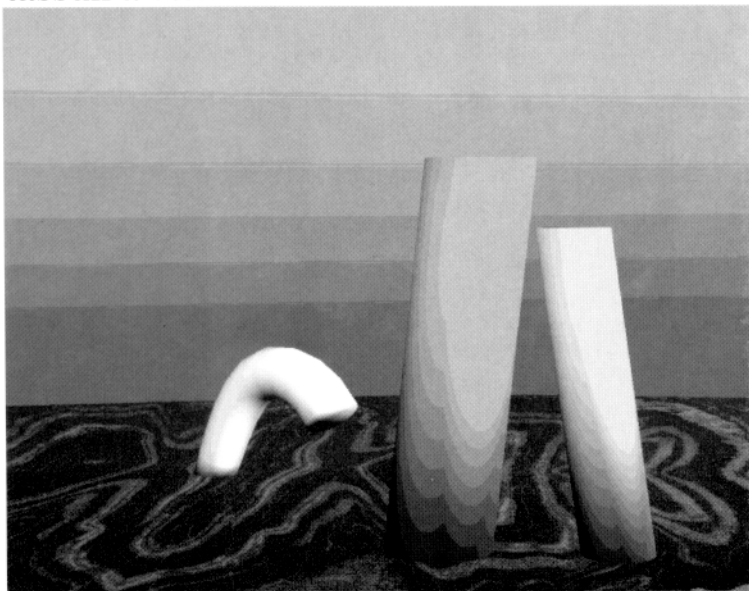
Ha anim vagy flick formátumú animációt készítenk, némileg más a helyzet. Ehhez nem használ az Imagine Movie fájlt, hanem mindig a képek sorrendjében készíti el az animációt. Az anim egyetlen fájlból áll, amely az alterv könyvtárában

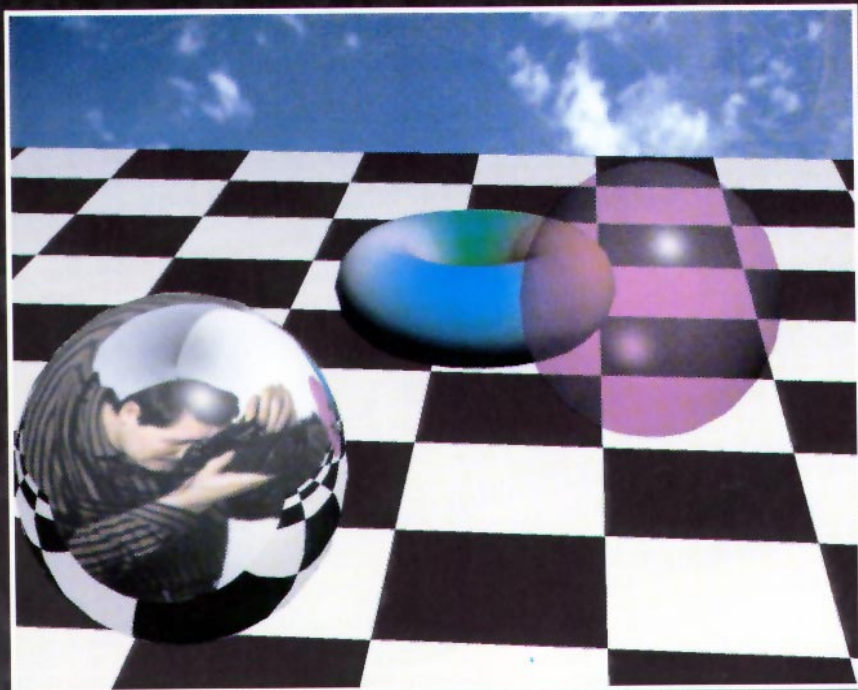
Animációs vezérlőfájl

helyezkedik el anim néven (ha a Parameters for Rendering Subproject kérdezőben nem adunk meg más útvonalat). Ha nem felel meg az összefűzési sorrend, külső animáció fűző programot kell használni. Én Amigán az **Imagine FX** nevű 24 bites képfeldolgozó programot használom e célra, amely program Magyarországon a **Thalypexn**él vásárolható meg. Ez a program, amellett hogy rengeteg hasznos képmanipulációs funkcióval rendelkezik, AREXX-ből is vezérelhető, kényelmesen, egyszerűen készíthető vele bonyolult képsorrendű animáció is.

Az Anim-5 és Flick típusú animációk lejátszását sem az Imagine, sem a PlayIANM nem támogatja, ezeket más animációlejátszóval lehet megtekinteni.

Miután mindent megtanultunk az animációk összefűzéséről, térjünk vissza az Action editorba, folytassuk a vele való ismerkedést.





A képek életszerűségének növelésére használhatunk digitalizált fotókat. A fenti képen a Global Brush egy fotó volt, amely engem ábrázolt fényképezés közben. Ez a fotó tükröződik a gömbben, olyan érzést keltve, mintha a kép egy fotó lenne, ugyanakkor az is látszik, hogy a kockás abroszon lévő tárgyak nem valódiak.

A Global Brush és a Backdrop Picture akár animálható is, ezáltal animációkban is létrehozhatunk ilyen vegyes hatást.

További lehetőség a valódi és a számítógépes kép keverésére a Genlock Sky, aminek használatával létrehozott képek, vagy azokból készített animációk háttérébe genlockkal video jelet keverhetünk. Ebben az esetben a háttér helyén a videojel fog megjelenni.

Az Attributes-ben is bekapcsolható egy Genlock opció, ekkor a tárgy helyén jelenik meg a videojel.

Animációk készítése

A következő munkánk legyen egy fekete-fehér kockás gömb átalakítása piros-sárga pöttyössé. Nyiss egy új tervet és altervet, majd a Detail editorban készíts el egy gömböt, amit ments ki a két különböző attribútummal mondjuk gömb-1.iob és gömb-2.iob néven. Az Action editorban állítsd be az animáció hosszát 20 kockára. A Globals-ban a háttérret állítsd be tetszőlegesen.

Menj a (new) sorba, és az Add-dal tegyél felsorolójelet az első képkocka alá. Válaszd a Normal objec-et és add meg tárgyként a gömb-1.iob-ot. A megjelenő Object File Info kéredezőben a Start és End Frame, valamint a Filename gondolom egyértelmű.

A **"Number of cycles to perform"**, az **"Initial cycle phase"** input mezők és a **"Reverse cycle motion"** kapcsoló a Cycle editorban készült, "önjáró", előre beállított mozgásokat automatikusan végző cycle objectekkel kapcsolatosak. Ismertetésükre a nevezett editor megtárgyalása után kerül sor.

A Transition frame count a Globals-nál ismertetettel azonos, az átalakulás hosszát beállító input mező. Ide most írd nullát. (Mivel a gömb-1.iob csak az első képkockában szerepel, amúgy se lehetne mást írni.)

Ugyan ehhez a színészhez a 2-20 képkockák közé töltsd be a gömb-2.iob-ot. Ennek a Transition frame count-ja legyen 18.

Kell még a színpadra legalább egy lámpa is, aminek elhelyezése már nem okozhat gondot.

Mivel nem valószínű, hogy elsőre sikerülne a pozíció és alignmet értékek beírásával pontosan összeállítani a színpadot, Save Changes után lépj át a Stage-be és helyezd el a tárgyakat az első kocká-

Animációkészítése

ban. Amikor egynél több képkockából álló terven dolgozol, az Action editorból a Stage-be átlépve egy input mező jelenik meg, **Frame Number** néven. Itt választhatod meg, hogy a Stage editor melyik képkocka állását mutassa.

Nem volt még szó a **Frame** menü pontjairól. Ezek segítségével mozoghatunk az egyes képkockák között, a Stage editorban.

A **First** az animáció legelső kockájára állítja az editort, míg a **Last** az utolsóra. A **Next** és a **Prev** az aktuális utáni vagy előtti kockára áll, míg a **Goto** egy tetszőlegesen kiválasztottra.

Ertelemszerűen nem lehet a Prev és Next segítségével az első képkocka elé, vagy az utolsó után ugrani.

A Camera View bekapcsolása után állítsd be a kamerát, hogy a gömb majdnem betöltse a képet. Használd a Camera (Re)Track-ot is.

A fényforrást vidd a kamera mögé, annál valamivel magasabbra.

Azok a beállítások, amiket a Stage editorban végzünk, a Save Changes hatására elmentődnek az Action editor felsoroló jeleiben. Természetesen csak azok a paraméterek tudnak elmentődni, amelyekhez az adott képkockáknál van felsorolójel. Ezért is fontos, hogy az első képkockában állítsd most össze a színpadot, mivel csak itt vannak pozíció és alignment felsoroló jelek.

Ha elkészült a beállítás, ne felejtse el kimenteni a változtatásokat! A képek kiszámoltatását és az animáció összeállítását rádbízom.

Tárgyak mozgatása egyenes mentén

Tárgyak mozgatása egyenes mentén

Az idáig tanultak alapján már mindenki be tudja rendezni a színpadot. Most próbáljuk meg mozgatni a színészeket. A darab, amit rendezünk legyen két szereplős, egy kocka és egy gömb adja elő. Nyiss egy új tervet, majd hozd létre a színészeket.

Állítsd be az animáció hosszát 20 képkockákra, és tedd fel a színre 1-20-ig a kockát.

A Stage-ben a kamerát úgy állítsd be, hogy a kocka tárgy középre essen, de ne töltsse ki a képeknek csak egy negyedét. A gömbünk ez előtt fog elhaladni balról-jobbra.

A Stage editorban az első kockában töltsd be a gömböt, és állítsd be úgy, hogy a kép bal szélén, a kocka előtt legyen. A Stage editorban betöltött tárgyak az aktuális kockától az animáció végéig bekerülnek az Action-ba. A pozíció, alignment és egyéb fesorolójelek csak az adott kockába íródnak be tehát a betöltött tárgy minden tulajdonsága változatlan marad az animáció alatt, hacsak később nem változtatjuk meg.

A golyót szeretnénk átmozgatni a jobb oldalra, a pozícióját tehát meg kell változtatni a 2-20 kockák között. Ezt megtehetjük mind az Action, mind a Stage editorokból.

Lássuk először, hogyan történik ez az Action-ból. Add módban tegyél egy fesorolójellet a gömb Position sorában a 2-20 kockák közé. A megjelenő **Specify Type** kérdésben válaszd a **Tween Position**-t.

Megjelenik a **Position Info** kérdéső, ahol a gömb végső pozícióját kell megadni. A tárgy a kijelölt animációs rész alatt elmozog ebbe a végpozícióba.

Ha azt szeretnénk, hogy a golyó az első öt kockában álljon a helyén, majd a hatodikban ugrás-

Animációk készítése

szerűen más pozícióba kerüljön, csak az első és a hatodik képkockákhoz kell felsorolójelet tenni, benne a pozíciókkal. A többi kép alatt nem lesz felsorolójel, mert ott nem változik a pozíció.

A legtöbb esetben nem tudjuk számszerűen azt a pozíciót, ahová a tárgyat mozgatni szeretnénk, ezért a beállítást a Stage editorból kell elvégezni.

Hagyd meg a Position Info-ban az értékeket, úgy klikkelj a OK-ra, majd a Save Changes után lépj vissza a Stage editorba, a 20. kockába. Állítsd be a gömböt a kép jobb oldalára. A Save Changes használata után az új pozíció bekerül a 2-20. kockák között húzódó felsorolójel End Frame Position Values input mezőbe, erről magad is meggyőződhetsz.

Ezzel kész az animációs beállítás, a gömb balról jobbra átmegy a kocka előtt.

Animációs előzetes

Az egyes képkockák kiszámoltatás előtt jó volna meggyőződni arról, hogy tényleg úgy zajlik-e le a folyamat, mint ahogy elképzeltük, nehogy csak a hosszadalmas kép és animációkészítés után derüljön ki az eseteleges hiba. Erre a Stage editor kínál egy gyors és kényelmes megoldást, az animációs előzetes készítest. Ez azt jelenti, hogy az animációt egy kis felbontású drótvázás megjelenítési formában elkészíthetjük, ami elegendő arra, hogy a folyamat helyességét ellenőrizni tudjuk.

Ezt az előzetes animációt a **Stage editor Animate** menüjén keresztül tudjuk elkészíteni és megtekinteni.

A **Make** pont alkalmazása után megjelenik egy olyan kérdező, mint a Project editorban a Range hatására. Feladata ugyan az, kijelöli az animációhoz

Animáció a Stage editorból

elkészítendő kockák tartományát és lépésközét. Ha ezt beállítottuk, elindul a generálás.

Az elkészült előzetes animációt a **Play Once** vagy **Play Loop** menüpontokkal megtekinthetjük azt. Az animáció lejátszását egy **Animation Controler**-rel irányíthatjuk. A **Stop** gomb megállítja a lejátszást, a **Step** egy kockát lép előre, a **Back** pedig vissza. Utóbbi két gomb csak a Stop után hatásos.

A **Play** a Stop-pal leállított animációt indítja újra. A lejátszás sebességét a **Slow-Fast** feletti tolókával tudjuk szabályozni. A Slow felé mozdítva a gombot lassul a lejátszás.

A **Rewind** hatására az animáció újra indul az elejéről. Kilépni a **Quit**tal lehet. A **Play Big** pontot kipipálva az animációt teljes képernyős méretben tekinthetjük meg.

Ha kiléptünk az animációból, az még a memóriában marad, később ismét megtekinthetjük. Ha szükség van az általa lefoglalt memóriára, a **Free Ram** törli az animációt a memóriából, felszabadítva az általa lefoglalt területet.

Az előzetes animáció mindig a perspektíva ablakban látható kép alapján készül, annak megjelenítési módjában (Wireframe vagy Solid.) Ha tehát a beállításnak megfelelő animációt akarunk létrehozni, a Camera View-et be kell kapcsolni! Lámpa nem kell ehhez a preview-hoz.

Animáció a Stage editorból

Mostmár akár összetett, egyenes szakaszokból álló mozgásokat is létre tudunk hozni, de az Action editorba folyton átkapcsolni kissé nehézkes, próbáljunk találni valami megoldást ennek kiküszöbölésére.

Lehet egy módszer, hogy a várható töréspontok

Animációk készítése

közé előre kirakjuk a felsoroló jeleket, de ennél tudok jobbat. Hogy egészben lásd a folyamatot, töröld le a gömböt, majd az első kockába állva töltsd be újra a Stage editorban. Mozgasd el a bal oldali pozícióba. Most menj a huszadik kockába és vidd a jobb oldalra a golyót. Mivel itt nincs felsoroló jel, hiába a Save Changes, nem marad itt a tárgy.

Aktiváld az **Object** menüből a **Position Bar** pontot, miközben a gömb van kiválasztva. Ennek hatására a legutolsó felsoroló jelet követő kockától (jelen esetben a 2.) az aktuális kockáig egy új felsorolójel jön létre az Action editorban, amelybe a végső pozíció értékéhez beíródik a tárgy jelenlegi pozíciója. Erről megbizonyosodhatsz ha átlépsz az Actionba, vagy elkészíted az animációs előzetest.

Mozgatás út mentén

Így már könnyebb több töréspontból álló mozgáso-rozatot előállítani, de még mindig bonyolult, ha egy görbe vonal mentén akarjuk mozgatni a tárgyat.

Ennek megkönnyítésére használhatjuk az utakat, amelyek készítéséről már többször is volt szó.

Mozgassuk most a golyót egy S alakú pályán. Ehhez az utat készítse el mindenki önmaga. A First point helyzete fontos, mivel innen indul a tárgy.

Az utat töltsd be az első kockában, így végig jelen lesz. Állítsd be úgy, hogy teljes egészében látható legyen, és a gömb tengelyének magasságában helyezkedjen el. Ez nem egyszerű feladat, mivel a perspektíva ablakban az út nem látható. Megkönnyítheted a dolgot, ha a végpontokba ideiglenesen valami egyszerű tárgyat, mondjuk egy-egy gömböt teszel. A beállítás után ezek a segédobjektumok törölhetők.

Mozgatás út mentén

A mozgatandó gömb pozíciója lényegtelen, azt úgyis az út fogja meghatározni.

Ha mindezzel megvagy, Save Changes után lépj át az Action editorba. Töröld le a gömb pozíció felsorolójelét, majd adj újat az 1-20. kockák közé. A Specify Type-ban válaszd a **Follow Path**-t, az útkövetést.

A tárgy által követendő út nevét a Path Name input mezőben kell megadni. Ha elfelejtetted volna megnevezni, ez legyen **PATH** (ez az út standard neve).

A tárgy az úton haladva nem csak egyenletes, hanem gyorsuló vagy lassuló mozgást is végezhet. Ennek beállítására szolgál az alsó négy input mező. A **Starting** és az **Ending Speed** a kezdeti és befejező sebesség Imagine egység/képkocka értékben.

A **(De)Acceleration** és az **(Ac)Deceleration** frames annak értéke, hogy hány képkocka alatt menjen végbe a gyorsulás vagy a lassulás. Ha ez nulla, a program egyenletes sebességgel számol. Ez így kissé zavaros, nézzünk rá egy egyszerű példát.

Egy tárgy 300 egység hosszú úton halad, 30 kocka hosszú animációs szakaszban. A normál sebessége tehát 10 egység/képkocka. Legyen a Starting speed 1 egység/képkocka, az Ending speed, pedig 5 egység/képkocka, a (De)Acceleration és az (Ac)Deceleration 10-10 kocka. A kérdés hogyan alakul a tárgy sebessége? Az első tíz kocka alatt egészen lassú sebességről felgyorsul, tíz kockányit halad egyenletesen, majd az utolsó tíz képkocka alatt fele sebességre lassít.

Ha a kezdeti vagy a befejező sebesség nagyobb mint az átlagos, a tárgy kezdetben lassul, vagy végén "hajráz".

Animációk készítése

Mozgó tárgy követése

Maradjunk most az egyszerű esetnél, mozogjon a gömb az úton egyenletes sebességgel. Egyszerre több tárgy is mozoghat ugyan azon az úton, egymással akár szinkronban, akár sem. Egy azonban közös lesz bennük, mind ugyan abba az irányba mennek.

Fejlesszük tovább az animációnkat, próbáljuk meg a kamerával követni a golyót, miközben az úton kanyarog. Egy megoldás máris kínálkozik, minden képkockában a Camera (Re)Track segítségével a gömbre irányítjuk a kamerát.

Ez túl nagy munka, felejtjük is el. Tudok jobbat, kettőt is.

Az egyik, hogy a kamera alignment-jének változtatásával elfordítjuk azt. Ezt hasonló módon lehet megtenni, mint a pozícióváltoztatást. Most is segítségünkre van a Stage editorban egy opció, az **Object menü Alignment Bar**-ja. Menj az első kockákba, állítsd rá a kamerát a gömbre, majd menj a 20-ba.

Válaszd ki a kamerát és használd az Alignment Bar-t. A 2.-20. kockák közé létrejön a kamera alignment felsoroló jele, amibe mostmár beírhatjuk az új értékeket. A Camera (Re)Track-vel állj rá a gömbre és mentsd ki a változtatásokat. Ezzel elérhetjük, hogy miközben a golyó végighalad az úton, a kamera utánafordul.

Az iménti módszerrel bárhová elfordíthatjuk a kamerát az animáció alatt, de mi van akkor, ha egy bonyolult három dimenziós mozgást végző tárgyat kell követni, miközben esetleg maga a kamera is mozog? Gyanítom, hogy így ritkán lesz a követendő tárgy a látótér közepén.

Két módszert, igérem, a kameramozgatásra és a második megoldást is jelent a felvetett problémára.

Mozgó tárgy követése

Lépj át az Action editorba, a következő kamerairányítás csak innen használható. Töröld le a kamera Alignment jeleit és adj egy újat a teljes animációs hosszra.

A **Specify Type** kérdésében három lehetőség közül lehet választani. A **Tween Alignment**-et már ismerjük, most töröld le a felsoroló jeleit. Ezzel a kamerát, de bármelyi tárgyat is a megadott szögbe lehet fordítani.

Az **Align to Path** kapcsoló használatának akkor van értelme, ha a tárgy pozíciója utat követ. Ekkor az Align to Path beállításának tartalma alatt a tárgy együtt fordul az úttal, mint például egy autó. Ebben a kérdésében egyetlen kapcsolót találsz, a Keep Y Horizontal-t. Ezt bekapcsolva a tárgy miközben együtt fordul az úton, Y orientációját végig megtartja, azaz nem csavarodik az úttal.

Az Align to Path eljárása teljesen azonos a Detail editorban az út mentén történő kinyomásnál részletezettel.

Amit mi keresünk, az a **Track to Object** mögött rejtőzik, klikkelj erre a kapcsolóra. A megjelenő kérdésében az Object Name input mezőben kell megadni azt a tárgyat, amelyre ez irányuljon. Írd most ide a gömb Actionban, vagy a Find by Name-ben szereplő nevét.

Az **Initial** és a **Final Y Rotation** a tárgy kezdeti és végső Y körül elfordulását jelenti.

A Track to Object-tel irányított tárgy Y tengelyének pozitív iránya néz a célként kijelölt tárgy tengelypontjára. Ezt nem csak a kameránál, hanem irányított lámpánál vagy bármilyen tárgynál, például egy repülőre kilőtt nyomkövető rakétának is használni lehet.

Ha a Track to Object be van állítva, a Stage editorban a Camera (Re)Track használatakor nem jelenik

Animációk készítése

meg kérdező, hanem a kamera nézőiránya a célként megjelölt tárgyra kerül. Ha a célt elmozgatjuk, a kamera állása nem frissítődik automatikusan, csak az említett menüpont használatakor.

Mire jó a követő tárgy Y körüli elforgatása? Például a kamerát a repülőgép fülkéjébe téve és az előtte haladó tárgyra irányítva megoldható, hogy amikor a repülőgép orsózik, a kamera valós, forgó képet közvetítsen.

Van úgy, hogy a kamerát nem közvetlenül egy tárgyra szeretnénk irányítani. Ekkor tesz jó szolgálatot az egyszerű Axis, amelyet felhasználhatunk a kamera céljának, akár mozgatva is, a végső képen mégsem lehet látni.

Természetesen az Axiist nem csak a kamera, hanem a lámpa, vagy bármilyen irányított orientációjú tárgy céljaként is használhatjuk.

Kamera forgatása

Most készítsünk egy olyan animációt, amelyben a kamera a függőleges tengelye körül tesz egy fordulatot, körbenéz a világban. A kamerának a kezdeti alignmentje legyen 0,0,0. Ugyan ez lesz a végső is.

Ha az alignment jel megosztásával a második, 2-20 kockák közötti részben is 0,0,0-t írunk, nem történik semmi, a kamera nem fordul egy fokot sem, nemhogy 360-at. Akkor mit tegyünk? Próbáljuk három felé osztani a fordulást, az első kocka maradjon 0,0,0-n, a 2-10 között fordítsuk el a Z körül 180°-t, majd a 11-20 között vissza nullára. Úgy gondolnánk most jó, de merre fogja megtenni a fél fordulatokat a kamera, jobbra vagy balra? Ezt nem tudjuk meghatározni.

Az alignment befolyásolásával történő forgatáskor a

Horgonypontok

tárgy a két irány közötti rövidebb íven fog elfordulni. Ahhoz, hogy a körülfordulás irányát befolyásolni tudjuk 180° -nál kisebb egységekre kell felbontani a körbefordulást, mondjuk 120° -okra. Az első jel egy kocka alatt lesz, 0° , a másodikban 120° , a harmadikban 240° , majd a negyedikben ismét 0° lesz az alignment értéke. Ha fordított irányban akarunk forgatni, a második felsorolójelbe kel a 240° -et írni, a harmadikba pedig a 120° -at.

Persze nem minden esetben lehet egyenlően elosztani a körbefordulást, például teljes fordulatot 10 kocka alatt téve. Ekkor úgy kell felosztani az egyes részfordulatokat, hogy mindegyik kockára azonos méretekű elfordulás jusson, jelen esetben 36° , és sehol sem legyen az elfordulás 180° , vagy több.

Miután az alignmentet alaposan kiveséztük, hozzáfoghatunk a következő feladathoz, változtassuk meg az animáció alatt egy tárgy méretét. Ehhez nem kell két különböző méretű tárgy között morfózist készíteni, elég a Size felsorolójellel manipulálni. A Stage editorban ezt az Object menü **Size Bar** pontja segíti. A beállítás elvégzése teljesen azonos a Position és Alignment-nél ismertettekkel, nem is részletezem külön. Elég annyi, hogy a kérdezőjében a tárgy három tengely iránti mérete van, ami összetett tárgynál nem mindig azonos a pontos mérettel, de arányaiban változtatva a kívánt hatást érhetjük el.

Horgonypontok

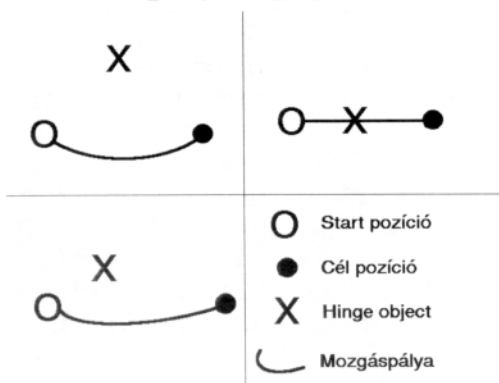
Térjünk vissza a gömbgurítós animációinkhoz. Állítsd be úgy a színpadot, hogy a golyó ne útmentén, hanem két végpozíció között, egyenes vonalban mozogjon, a kocka előtt jó három gömbnyi távolságban. A kamerát helyezd el úgy, hogy felülről

Animációk készítése

tekintsen le a tárgyakra, de ne kövesse egyiket sem.

Készítsd el az előzetes animációt. A golyó a várakozásnak megfelelően egyenes vonalban halad. Most lépj át az Action editorba és a gömb **Hinge** feliratú sorába tegyél felsorolójelet az 1-20. kockák közé. Egy kérdező jelenik meg, a **Hinge Object** input mezőbe írd be a kocka kincstári nevét. Készítsd el ismét az előzetes animációt. Ha kész, nézd meg és csodálkozz, a golyó íven halad, mintegy kikerüli a Hinge object-jét, a kockát.

A Hinge szó szerinti fordításban horgonyt jelent. Az Imagine-ben egy tárgy a Hinge object-ként kijelölt tárgyhöz horgonyzódva mozog. Ha a kiindulási és a végponttól egyenlő távolságra van a Hinge object, a mozgás köríven történik, egyébként ellipszis mentén. Ha a horgonytárgy pont a mozgó tárgy két végpontja



közötti egyenesen helyezkedik el, a horgonyzás hatástalan. A pontos útvonalat, amin a lehorgonyzott tárgy haladni fog, nehéz meghatározni, ezért inkább Path-okat használunk a görbe vonalú mozgások vezérléséhez.

Animációs effektusok

Ezzel elérkeztünk az Action és Stage editorok utolsó részéhez, a speciális effektek használatához. Ezek az effektek külső programmodulok, amelyeket az Action editorból használhatunk föl. Hozzárendelve

Animációs effektusok

valamelyiket (egyszerre akár kettőt is) egy tárgyhoz, annak megváltoztatják legtöbbször a geometriáját, de néha a felületi tulajdonságait is. Ilyen effekt lehet például a robbanás, minek hatására a tárgy egy adott animációs rész alatt az alkotó felületelemeire esik szét.

Az effekteteket általában a program könyvtárán belül az Effects könyvtárban találjuk. Az Imagine 2.0-s változatában nyolc előre elkészített effektus található, de bizonyos programozási ismeret és az effekték működési leírásának birokában újabbak és írhatók. Megjegyzem utólag írt textúrával már talákoztam, effektel még nem.

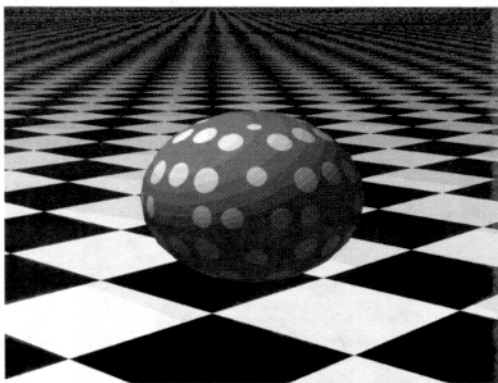
Az effekték az **F/X 1** és **F/X 2** sorokba tett felsorolójelekkel rendelhetjük a tárgyakhoz. Legegyszerűbb lesz, ha ABC sorrendben mindegyiket kipróbáljuk.

Mivel a hatások a tárgyakon jönnek létre, függetlenül a világban elfoglalt pozíciójukból, a tengelyek alatt mindig azok saját tengelyét értjük.

Az első effektus a **Boing**. Ez valamely, egyszerre csak az egyik tengely mentén összenyomja vagy széthúzza a tárgyat, majd visszaállítja az eredeti állapotába, olyan hatást keltve, mint például egy pattanó gumilabda.

Azt, hogy az összenyomódás-széthúzódás melyik tengely mentén menjen végbe, az **X,Y,Z Axis** kapcsolókkal lehet kiválasztani.

A **Squash to ...** kapcsolókkal azt tudjuk szabályozni, hogy a torzulás a



Animációk készítése

tárgy melyik részén menjen végbe. A **Center** hatására a tárgy a tengelyközéppontra zsugorodik, vagy tágul, azonos hatást keltve, mint a **Scale**.

A **+** vagy **- (X Y or Z)** bekapcsolása után a tárgynak a tengelyközéppontból a kiválasztott tengelyen pozitív vagy negatív irányba eső része fog torzulni.

A **Shrink Factor** az összenyomódás vagy a széthúzódás mértéke. Ha ez egynél kisebb, a tárgy először zsugorodik, majd visszaáll eredeti méretére, ha nagyobb mint egy, először kitágul, majd visszahúzódik. A **0,75 Shrink Factor** az jelenti, hogy a tárgy az animációs rész első felében háromnegyedére zsugorodik az egyik tengely mentén, majd a második, ugyanekkora szakaszon nyúlik vissza eredeti méretére.

A **Number of Times** input mezőn keresztül tudjuk megadni, hogy a kijelölt kockák alatt a tárgy hány-szor hajtsa végre a torzulásciklust. Ha ez kettő, a tárgy kétszer zsugorodik-tágul.

A következő effekt az **Explode**, azaz robbanás. A tárgyat három féle módon robbanthatjuk fel: gömb-szerűen, amikor a darabok egy középpontból, a tengelypontból széttartva repülnek, hengeresen amikor egy tengely körül sugárirányban távolodnak a darabok, és egyenes mentén, amikor a szétröppenő



darabok valamely tengely irányában haladnak. Ennek beállítására szolgálnak a **Spherical**, **Radial** és **Linear** kapcsolók.

Utóbbi két esetben tengelyt is kell választani az **X, Y, Z** kapcsolók valamelyikével.

Az **Explosion Distance** az a távolság, amelyre a szétröppenő darabok az effekt működése alatt eltávolodnak eredeti pozíciójuktól.

Az **Expansion Angle** a darabok emelkedési szöge a robbanás után.

A **Triangle Scaling** segítségével a darabok, miután kiváltak a tárgyból, megváltoztathatják méretüket. Az input mező értéke a felületelemek végső és eredeti méretének aránya. Mivel a méretváltozás a felületelemeken külön-külön megy végbe, a tárgy méretére ez nincs hatással. A Scale értéke sohasem lehet 0, így a darabokat teljesen eltüntetni nem lehet, csak a színész eltávolításával.

Lehetőség van arra is, hogy a darabok, miközben távolodnak, forogjonak is, véletlenszerű irányokban. A **Minimum** és **Maximum# of Rotation** a darabok által legalább és legfeljebb végzendő fordulatok száma. Minden darab e két érték közötti, véletlenül választott fordulatot végez, növelve a robbanás életszerűségét.

A **Random number seed** egy tetszőleges szám. Ezt a véletlenszám-generátor használja fel az előbbi funkció véletlen számának kiszámításához. Gyakorlatilag nem sok jelentősége van, értéke bármi lehet.

A kérdező alján található két kapcsoló. A **Return to original position**-t bekapcsolva a szétrobbanó darabok csak az animációs ciklus első felében távolodnak egymástól, félúton a folyamat megfordul és a darabok ismét összeállnak egy egész tárggyá.

A **Reverse explosion timing** bekapcsolása után a robbanás fordítva megy végbe, az első kockában lesznek leginkább szétszóródva a darabok, amelyek az utolsó képkockában összeállnak egy tárggyá. Ez az opció nem működik együtt a Return to original position-nal.

Az effektus már az első képkockán érezteti hatását,

Animációk készítése

amelyiken alkalmaztuk, csakúgy mint minden effekt, így már az első kockában, vagy megfordított robbanás esetén az utolsóban sem lesz a tárgy összefüggő egész. Ha erre szükség van, az effektel úgy kell kijelölni, hogy maradjon legalább egy kocka a színész szereplésében, amikor az effekt nem hat rá.

A **Fireworks** nagyon hasonlít az Explode-ra, a hatásában mindössze az a különbség, hogy számol egy Z irányú gravitációval, a szétrobbanó darabok íves pályán repülnek, amely a "föld" felé hajlik, a darabok visszahullanak a Ground-ra. A kérdező is alig különbözik az Explode kérdezőjétől. Az azonos nevű input mezők természetesen ugyan azok, mint az előbb, az újak jelentése a következő:

Distance to Fall by End - Annak a távolságnak a mértéke, amely után az eltávolodó darabok eredeti magasságukra hullanak vissza, átvittén tehát a gravitáció nagysága.

Make Faces Sparkle - Kapcsoló, melyet ha kiikszelünk, szikrázó hatást keltve színeződnek a széthulló darabok.

A következő effekt, a **Flash** igen egyszerű, kérdezőjében mindössze két input mezőt és két kapcsolót találunk. Az effektel a tárgyat villogóvá tehetjük, mint például egy rendőrautó szirénája. A villogás nem jár sem a tárgy elszíneződésével vagy fényvisszaverő képességének megváltozásával, se nem kezd fényforrásszerű tulajdonságokat mutatni. Mindössze annyi történik, hogy a tárgy **Bright** tulajdonsága, amit a egyébként a Detail editor Attributesében lehet beállítani, ciklikusan ki-be kapcsolódik. Az effekt független a tárgy saját Bright attribútumától.

Az **On Frames** azon képkockák száma egy villanó cikluson belül, amelyeken keresztül a tárgy Bright

tulajdonsága be van kapcsolva. Az **Off Fames** a kikapcsolt **Bright** képkockáinak száma.

A tárgy **Bright** tulajdonságának változása minden átmenet nélkül, hirtelen megy végbe.

Azt, hogy az effekt hatásának kezdetén a **Bright** milyen állapotban legyen, a **Start On - Start Off** kapcsolókkal lehet szabályozni. A **Start On** a bekapcsolt állapotú kezdést jelenti.

A **Grow** effekt paramétereit megvizsgálva, azok ismerősnek tűnnek. Megmondom honnan, a **Mold Extrude** funkciójából. Az effekt ugyan ezt a hatást hozza létre, de a tárgy kinyomása egy animáció alatt megy végbe. Lényeges különbség, hogy a **Grow** mindig **Path** mentén nyomja ki a tárgyat.

Hogy ne kelljen annyit lapozni, az egyes paramétereinek jelentését röviden megemlítem. Részletesebb leírást az említett **Extrude** funkciónál olvashatsz.

Figyelmesekek rögtön észrevehetik hogy nincs input mező az út megadására, pedig arról volt szó, hogy a **Grow** mindig út mentén nyomja ki a tárgyat. Nos a **Path** megadása némileg másként megy. A tárgyat és az utat a **Detail** editorban kell csoportba kötni, úgy hogy az út legyen a tárgy szülője. Ezt a csoportot kell azután a színpadra helyezni, és alkalmazni rá a **Grow** effektet. Az út és a tárgy egymáshoz viszonyított helyzete lényegtelen, a tárgy az úton a **First Point**-ből indulva nyomódik ki, csakúgy, mint az **Extrude**-nál.

Az **Y Rotation** a tárgynak a kinyomás közbeni elcsavarodása fokban.

Az **X, Z Scaling** a méretváltozás aránya.

A **X, Z Translate** a kinyomás közbeni eltolás a tárgy valamelyik tengelye mentén.

Az **Align Y to path direction** bekapcsolásával a tárgy **Y** tengelyének iránya is követi az utat.

Animációk készítése

A **Keep X in path s X-Y plane** bekapcsolásával elérjük, hogy a tárgy megtartja X tengelyének eredeti orientációját, nem csavarodik együtt az úttal.

A **Mirror ends** a kinyomott tárgy utolsó szegmensét tükrözi.

A **Time Reversed** hatására a kinyomás folyamata fordítva zajlik, le a teljesen kinyomott tárgy zsugorodik vissza a kiindulási méretre.

A **Ripple** effekttel hullámozó felületet hozhatunk létre. A hatás hasonlít a Waves textúrára, de itt az animáció alatt, folyamatos hullámozgás jön létre.



A hullámozás lehet egy pontból körkörösén szétterjedő, mint mikor egy kavics esik a vízbe, vagy egyenes mentén haladó, mint a szélfújta vízfelület, vagy a tenger hullámozása. E két lehetőség között a **Radial around Z** (Z körül körkörösén) és

a **Linear along X** (X mentén egyenesen) kapcsolókkal választhatunk.

A **Wavelength** a hullámhossz, vagyis a két hullámhegy közötti távolság Imagine egységben.

Z Amplitude a hullámok magassága szintén Imagine egységben. Értéke erősen függ a tárgy méretétől. Például egy 100 egység átmérőjű földgömbön lévő óceán 10 egység magas hullámzása már szökőárnak minősül, ellenben, ha a földgömb 1000 egység átmérőjű a 2 egység magas hullámverés szinte észrevehetetlen.

A **Travel Distance** az a távolság, amelyre a hullámok eljutnak, mielőtt elhalnának. Minél sűrűbb a hullámzó közeg, a hullámok annál hamarabb halnak el. Például egy homokba csapódó tárgy átmérőjének alig kétszeresére hatoló hullámzást hoz létre, míg vízben a tárgy által keltett hullámzás az átmérőjének több tízszeresére is eljut.

A **Ripple Count** értéke adja meg, hogy hány hullám induljon el a tárgyon. Egy széllobogtatta zászlón 10-15 hullám is végighalad, míg egy kis tócsába eső kavics nyomán jó ha 3-4 hullám indul meg. Természetesen minden hullám ugyan abból a pontból ered, körkörös hullámzás esetén mindig a Z tengelyből, egyenes hullámverés esetén a Z-Y síkból.

Mivel a Ripple effekt a tárgy pontjainak átrendezésével hozza létre a hullámzást, minél több pontból áll a tárgy, annál szebb lesz a hatás. Például egy 2*2 szegmensből álló sík hullámozgatása nem hoz látványos eredményt.

A **Rotate2.0** igen egyszerű dolgot végez, valamely tengely (egyszerre csak egy) körül megforgatja a tárgyat. Ezt a hatást tulajdon képpen az Alignment paraméterrel is el tudnánk érni, de ez jóval kényelmesebb, egyszerűbb módszer. Egy effektel folyamatosan több fordulatot is végezhet a tárgy.

A forgatás tengelyét az **X,Y,Z Axis** kapcsolókkal tudjuk kiválasztani. Az elforgatás szögéért a **Degrees** input mező felelős. A forgás az adott tengely körül pozitív irányban történik. Ha ellenkező irányban akarjuk a forgatást végrehajtani, a szögértékhez negatív számot kell írni.

A **Tumble** effekt hasonló a Rotate-hez, de ennél a forgás véletlenszerűen kiválasztott tengely körül, véletlenül meghatározott fordulattal történik. Több

Animációk készítése

tárgyat a Tumble segítségével megforgatva igen kaotikus animációt hozhatunk létre. "Ez mire jó?" kérdezik sokan, nem kis élel. Igen is nagyon hasznos. Én a tárgyak robbantásánál szoktam használni. Az Explode effekt hatására létrejövő háromszögű darabkák, közelebről felvéve a robbanást, nem mutatnak igazán, jól. Helyete inkább több apró darabból hozom létre a tárgyat, majd ezeket a darabokat pozícióváltoztatással eltávolítom egymástól, miközben mindegyikre alkalmazom a Tumble effektet. A robbanás életszerűségének növelésére a tárgy eredeti helyére egy gyorsan növekvő, alakját változtató, színes ködszerű tárgyat teszek. Igaz ez a módszer nehezkesebb mint az Explode, de sokkal szebb eredményt ad.

Másik felhasználási területe a Tumble-nek a meteorzápor, ahol az egyes meteoritok rendezetlenül forognak.

A **Random Axis** kapcsoló bekapcsolásával a forgás tengelye véletlenszerűen választódik ki, a **Z axis** hatására a forgás mindenképpen a Z tengely körül történik.

A **Minimum# of rotations** a tárgy által minimálisan megteendő fordulatok száma. Ez fordulatot jelent és nem elfordulási szöget!

A **Maximum# of rotations** a maximálisan metett fordulatok száma. A fordulatok pontos száma e két érték között véletlenül választódik ki.

A **Random number seed** egy tetszőleges szám, a véletlengenerátor használja, érteke bármi lehet.

Ezzel befejeztük a textúrák, az Action editor és a Stage editor ismertetését. A Cycle editorral való alapos megismerledés után még vissza fogunk térni az Action-be, hogy megtanuljuk a ciklikus tárgyak felhasználásának módját.

Cycle editor

A Cycle editor az Imagine egy olyan újítása, amelyre más, személyi számítógépeken futó ray-tracer program mind a mai napig nem képes. Ebben az editorban ciklikusan mozgó tárgyakat hozhatunk létre, amelyek a színpadra kerülve az előre beállított mozgássorozataikat automatikusan végzik. Klasszikus példa erre a sétáló ember esete. A tárgyat megtervezve, az egyes mozgásokat, mint lépés, kézmozgás, csípőmozgás, stb beállítva, a tárgy az animációban automatikusan sétálni fog.

Ez már önmagában is nagy dolog, de ehhez járul még, hogy a mozgásokat kulcskockákkal tudjuk beállítani, vagyis elegendő egy mozgássor két végállapotát megadni, a program kiszámolja a köztes fázisokat. Ezt az eljárást Cell animációnak, kulcskockás animációnak nevezik.

Az előbb említett sétáló mozgáson kívül, bármilyen ciklikusan ismétlődő mozgássorozatot létrehozhatunk a Cycle editorban, mint például a repülőgép forgó légcsavarja, az ingaóra ingája, a gőzmozdony tolattyúja, stb.

A Cycle editorban csak a mozgásfázisokat tudjuk megtervezni, a mozgó tárgyak alkotó elemeit tehát előre el kell készíteni a Detail vagy a Forms editorban.

Az első tárgy, amit a Cycle editorban elkészítünk legyen egy inga. Ezt az ingát a lehető legegyszerűbb elemekből állítsuk össze, két csőből. Az egyik lesz az inga állványa, a másik a karja. Tulajdonképpen elég egyetlen csövet megtervezni, azt fel tudjuk használni mindkét helyre.

A méret nem fontos, a tengely pozíciójára és méretére azonban van néhány fontos megkötés. A tengelypont helyezkedjen el a cső egyik végén, a Z

Cycle editor

tengely pedig haladjon a cső hosszában és pont olyan hosszú legyen mint maga a cső.

Hogy ez miért fontos? A Cycle object úgy épül fel, mint a csoportok, minden tárgynak, kivéve a csoport legfelső elemét, van egy szülője. Az egyes elemek csuklósan kapcsolódnak egymáshoz, minden gyermek Z tengelyének vége a szülője tengelypontjához kapcsolódik. A csuklópontot tehát a szülő tengelypontja és a gyermek Z tengelyének vége alkotja. Ahhoz, hogy az ingát alkotó csövek a végükkel illeszkedjenek a tengelypontot és a tengelyvéget a cső két végére kell helyezni! Most persze ez még zavarosnak tűnik, de nemsokára teljesen érthetővé válik.

Miután elkészült a cső, lépj át a Cycle editorba. Az editor elrendezése a szokásos négy ablakos. A három síknézetben eleve látható egy tengely. A Cycle editorban egyszerre csak egy cycle object szerkeszthető, a jelen lévő tengely ennek a csoportnak lesz az ősszülője.

A szerkesztő **Project** és **Display** menüi a szokásos menüpontokból állnak, részletezésüktől most tekintsünk el.

Az **Object** menüben mindössze három menüpont található. A **New** törli a szerkesztőt és elhelyez benne egy **Axis**-t. A **Load** ponttal tudunk cycle object-et, vagy előre összeállított csoportot betölteni. A **Save** ponttal csak cycle object formátumban lehet kimenteni a szerkesztett csoportot.

A **Pick** menü két pontra korlátozódik, a **Groups**-ra és az **Object**-re. A **Groups** módban egész csoportokat, a kijelölt tárgyat és a belőle eredő gyermeket tudjuk manipulálni, míg az **Objects**-ben csak a kijelölt tárgyat, függetlenül a csoportban elfoglalt helyüktől.

A **Mode** menü pontjai teljesen újak, tehát részletes magyarázatra szorulnak, ami nemsokára meg is történik.

A **Cell** menüben a ciklus kockái között való mozgást és ehhez kapcsolódó funkciókat megvalósító pontok kaptak helyet.

Az utolsó menü, az **Animate** teljesen ismerős a Stage editorból, a ciklusról készíthetünk előzetes animációt és tekinthetjük meg azt.

Mielőtt az egyes menüpontok jelentésével és használatával konkrétan is megismerkednénk, lássuk, hogyan épül föl egy ciklikus object mozgása. Ennek megértése fontos a Cycle editor eredményes használatához.

Azt már tudjuk, hogy egy cycle object Detail vagy Forms editorban elkészített tárgyakból áll, amelyeket csoportba kötünk. A tárgyak egymással csuklósan kapcsolódnak. A cycle object tartalmazza egyszer a csoport hierarchiáját, az azt alkotó tárgyakat, valamint az egyes tárgyak által végzett mozgások leírását. Egy elmozduló kart tehát nem kell kétszer eltárolni, elég a tárgy csoporton belüli pozícióját és az elmozdulás kulcspozícióit tárolni. Innen adódik az elnevezés, a kulcs kockás vagy cell animáció.

Vegyünk egy egyszerű esetet, egy ingát. Az inga csuklósan kapcsolódik a tartószerkezetéhez, például egy faliórához. Ahhoz, hogy az inga mozgását modellezni tudjunk elég ismerni az inga lengésének két végpozícióját, és hogy a két pozíció közötti utat hány képkocka alatt teszi meg. Tegyük fel, hogy az inga az egyik pozícióból a másikba 10 kocka alatt lendül át. Egy teljes lengésciklus tehát három kulcskockából és 20 animációs kockából, frame-ből áll.

Az első kulcskocka a nulladik. Ez a kiindulási helyzet, az inga legyen a bal oldali végállásban. A második kulcskocka tízzel arrébb van, tehát a 9.

Ebben az inga a jobb oldali végpozícióban helyezkedik el. A harmadik kulcskocka az utolsó, a 19., amelyben az inga a bal oldali végállásba kerül. A folyamat ciklikusan ismétlődik, az animáció 21. kockájában ismét a ciklus 0. kockája, az első kulcspozíció következik.

Az előző példa kissé sántít, mivel a bal oldali kulcspozícióban két frame-nyit tartózkodik, a 19.-ben és az azt követő 0.-ban. Ezzel szemben a jobb oldalon csak egy frame-nyit tartózkodik az inga, a második kulcspozícióban, a 9. képkockában. A megoldás, hogy a 10. képkockát is kulcspozícióvá kell tenni, amelyben az inga a jobb szélső helyzetben marad. Így már jó lesz a lengés, mivel az inga a végpozícióban megtorpan, úgy változtat lengési irányt.

A köztes képkockákat a program automatikusan generálja, amikor a ciklikus tárgyat felhasználjuk. (A példa a másik lábára is sánta, mivel a ciklikus tárgy a kulcspozíciók között egyenletes sebességű mozgást végez, míg gyakorlatban az inga sebessége folyamatosan változik, de ettől most tekintsünk el.)

Az, hogy a lengést hány frame-re állítjuk be, csak itt, a Cycle editorban a ciklus időarányainak beállításához fontos. Később, a tárgy felhasználásakor pontosan meghatározhatjuk, hogy a ciklus valójában hány képkocka alatt játszódjon le. Tegyük fel, a példában szereplő 20 képkockás ismétlési idejű lengést egy 40 kockás animációban használjuk fel úgy, hogy épp egyszer menjen le a ciklus. Ebben az esetben a 9. kulcskocka helyén lévő cella a 18. kockába helyeződik át. Az eredetileg 10. kockában lévő cella a 20. kockában lesz, míg a befejező cella a 19. kockából átkerül a 38.-ba. A ciklus így egyszer játszódik le, az eredetinel kétszer hosszabb időtartamban. Az inga végpozíciókban való tartózkodá-

sának ideje kettő frame-ről négy frame-re nyúlik, az egész animációs ciklus időzítése, arányai nem változnak.

Legyen egyelőre elég az elméleti fejtegetésből valósítsuk meg az ingát a gyakorlatban is. Ehhez először a két alkotórészt el kell helyezni a ciklusban.

Mivel a Cycle editorba való belépésünk óta még nem hajtottunk végre semmilyen műveletet, a szerkesztő a 0. kockán áll, amely egyben kulcspozíció is. Ezt jelöli az állapot sorban a 0. szám és az után álló csillag, ami a kulcskocka, más néven **Cell**, cella jele. Az **Add** jelenti hogy abban a módban vagyunk, melyben a ciklikus tárgy struktúrájához új elemeket adhatunk.

Tárgyat a ciklushoz adni csak a 0. kockában, az első kulcspozícióban lehet. A ciklikus tárgy a ciklus alatt nem változtathatja meg a szerkezetét, tehát nem adhatunk hozzá új tárgyat, nem cserélhetjük le valamely elemeit, nem változhat meg a csoportszerkezet sem. Ha a **Mode menü Delete** pontjának kiválasztása után valamely rombusz tengelypontjára klikkelünk, az a rombusz, a belőle származó összes elemmel együtt törlődik, a teljes ciklusból, tehát ezen a módon sem tudjuk a cikluson belül módosítani a szerkezetet.

Állítsd be a kurzor nyilakkal a front nézetet úgy, hogy az összülőt jelképező tengely a nézet felső részére kerüljön. Most vidd a kurzort a tengelypontjára, nyomd le az egér bal gombját, és a nyomva tartása közben húzd le az egeret kissé. Egy sárga rombusz rajzolódik ki, amely az inga első darabját jelképezi.

Most szembe találtuk magunkat a Cycle editor megjelenítésének egy különlegességével. A síknézetekben nem a konkrét alkotó tárgyakat látjuk, hanem az azokat helyettesítő rombuszokat. Ez az

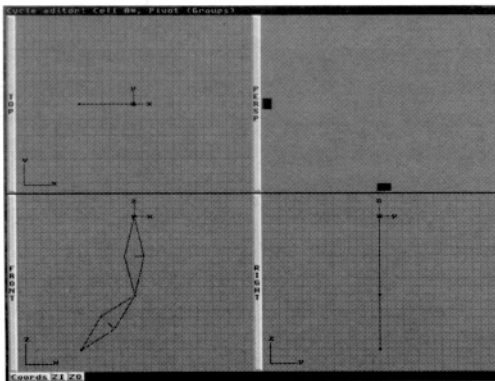
Detail editor

eljárás jelentősen gyorsítja képmegjelenítést és átláthatóbbá teszi a bonyolult felépítésű ciklikus tárgyakat. A rombuszokhoz egy rövidesen bemutató módon rendelhetjük hozzá a valós tárgyakat. Amíg a hozzárendelés történik meg, addig a perspektíva nézetben nem látunk semmit.

A rombusz úgy helyezkedik el, hogy a hozzá rendelendő tárgy Z tengelyének vége kapcsolódik az összülőt jelképező tengely origójához, míg a saját tengelypontja ott lesz, ahol az egér bal gombját elengedtük. A tárgy X tengelye a rombuszban keresztben helyezkedik el, ezt jelöli az a kis vonal a belsejében. Fontos tehát, hogy a rombuszok nem a tárgy, hanem csak a tengelye méretét jelölik. A kapcsolódó tárgyaknál mindig a szülő tengelypontjába illeszkedik a gyermeke Z tengelyének vége.

Például, ha a tárgy tengelypontja, vagy Z tengelyének vége kívül esik a tárgyon, azon az oldalon hézagos lesz a kapcsolat. Fordított esetben, amikor a tengely a tárgy belsejében van, a kapcsolódó tárgyak egymásba érnek.

Folytassuk az inga építését, helyezük el az inga szárát is. Mivel a csoportszerkezeben minden tárgynak (kivéve az összülőt) egy másik tárgyból kell



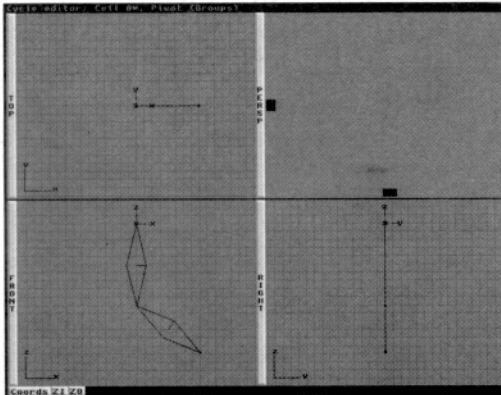
származni, a következő rombuszt csak két helyről indíthatjuk: a tengelyből, vagy az előbb lerakott rombusz tengelypontjából. Első esetben az új elem azonos hierarchikus magasságra kerül az előbb létrehozott ingaállvánnyal, míg a

második lehetőséget választva az inga állványához fog csatlakozni. Utóbbi az, amire szükségünk van.

A rombuszok mérete fontos, azok fogják meghatározni a hozzájuk rendelendő tárgy méretét. Ha a két rombusz nem egyforma, még ha ugyan azokat az elemeket rendeljük is egymáshoz, azok különböző méretűek lesznek. A tárgyak méretváltozása mindhárom tengely mentén azonos arányú.

A két rombusz elhelyezésével létrehoztuk az inga hierarchikus szerkezetét, mostmár beállíthatjuk a kulcspozíciókat.

Jelenleg a 0. kockában vagyunk, ami mindig kulcscocka, ez határozza meg a ciklikus kiinduló helyzetét. A **Mode** menüből válaszd ki a **Pivot**-ot. Ebben a módban a tárgyakat csatlakozó pontjaik körül tudjuk forgatni. Fogd meg az előlnézetben a második rombusz tenelypontját, vagyis azt a



csúcát, amelyben az egér gombját elengedted. Forgasd el a rombuszt balra, kb. 45°-os szögben. Ezzel beállítottuk a bal oldali végpozíciót. Most a **Cell** menü **Goto** pontjának segítségével menj a 9. frame-re, ahogy azt az elméleti részben megbeszéltük.

A ciklusunk most még csak a 0. kockából áll, hogyan mehetünk el akkor a 9.-be? Nagyon egyszerűen, amikor egy cikluson kívüli kockára ugrunk, a ciklus automatikusan kibővül addig a kockáig. Sőt, a program ilyenkor feltételezi, hogy ezért bővítettük a

Cycle editor

ciklust, hogy ide kulcskockát helyezünk, ezért ez automatikusan cellává válik. Az új cellában a tárgy helyzete a bővítés előtti utolsó celláéval azonos.

Ne felejtse el, hogy egy ciklus első és utolsó kockájának mindig kulcskockának kell lenni, mert csak így tudjuk a kezdő és befejező fázisokat definiálni.

Pivot módban maradvá, forgasd el az ingát a jobb oldali végállásba. Ezután bővítsd tovább a ciklust a 10. kockáig. Ebben a kulcspozícióban az inga még mindig a jobb szélső helyzetben van, ezért nem kell rajta változtatni.

Az utolsó cella a 19-es. Amikor ezt létrehozod, benne a tárgy a jobb oldali véghelyzetben van.

Hogyan pozicionáljuk az ingát pontosan a 0. kockában lévő bal oldali helyzetbe? A **Cell menü Copy From** pontjával. A megjelenő input mezőbe írd be annak a kockának a számát, ahonnan a pozíciót át akarod venni, jelen esetben tehát a 0-t. Az inga ezáltal ugyan abba a helyzetbe kerül, mint a 0. kockában, a mozgása folyamatos lesz.

A **Next Key** és a **Prev Key** pontjaival a **Cell menü**nek oda-vissza végignézhethetjük az egyes kulcskockákat. A **First** ponttal a ciklus elejére, a **Last**-tal pedig a végére ugorhatunk. A **Next** és a **Prev** pontokkal a ciklust folyamatában lapozhatjuk végig, megtekintve a köztes fázisokat is. A **Goto** segítségével természetesen nem csak kulcspozícióra, hanem köztes fázisokat is ugorhatunk. A ciklus felépítését segíti, hogy a kockák között lépegetve az előző vagy visszafelé lépve a következő kocka pozíciója halványan kirajzolódik. Kulcskockák közt mozogva az előző/következő cella rajzolódik ki halványan.

Tulajdonképpen az **Anim menü Make** pontjával már elkészíthetnénk a ciklusból az animációs előze-

test, de mivel az egyes elemekhez még nem rendeltünk konkrét tárgyat, nem látnánk semmit.

Menj a 0. kockába és kapcsolj át **Assign** módra. Ha most valamely rombusz tengelypontjára klikkelsz, egy fájl kérdező jelenik meg, amelyben egy tárgyat kiválasztva, az hozzárendelődik az adott rombuszhoz. A hozzárendelt tárgy képe megjelenik a perspektíva ablakban, láthatóvá téve a ciklikus tárgyat. Hozzárendelést csak a 0. kockában lehet létrehozni. Minden rombuszhoz egyszerre csak egy tárgy tartozhat. A tárgy csak önálló object lehet, csoport nem.

A hozzárendelés nem föltétlenül az utolsó mozzanata a ciklikus tárgy létrehozásának, akár már az első kulcskocka felépítése után végrehajthatjuk. Ebben az esetben a további kulcspozíciók kialakításakor már a perspektíva képét is láthatjuk a cycle objectnek.

Ha egy olyan rombuszhoz rendelünk tárgyat, amelyhez már van valami hozzárendelve, az új hozzárendelés felváltja a régit. Minden rombuszhoz csak egy tárgy lehet egyszerre rendelve.

A **DeAssign** segítségével a tárgyhozzárendelést szüntethetjük meg. Ebben a módban, ha ráklikkelünk valamely rombusz tengelypontjára, akkor ahhoz a rombuszhoz nem lesz object rendelve.

Mentsd ki az ingát az **Object menü Save** pontjával, majd a **New** alkalmazása után készítsünk egy más típusú cycle objec-et, egy helikopter rotort. Az egyszerűség kedvéért ez is az ingát alkotó csőből fog felépülni, tehát nem kell a Detail-ban új elemeket létrehozni.

Az előlnézetben az ősszülőből felfelé hozz létre egy szegmenst, ez lesz a rotor tengelye. Most menj át felülnézetbe és csatlakoztass a tengelyhez négy rotortlapátot. Ha a rotortengelyt pont függőlegesre

Cycle editor

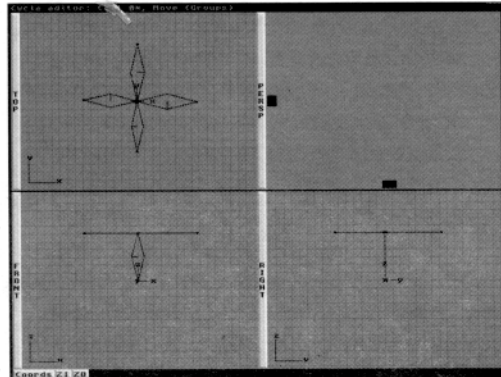
készítetted, nehéz a felülnézetben, ahol a rombusz két vége és az Axis egybe esik, a rotortengely felső végéhez, a tengelypontjához illeszteni az új elemeket. Ennek elkerülésére a rotor tengelyét jelképező rombuszt ferde-re készítsd el, majd a rotorlapátok elhelyezése után egyenesítsd ki.

A rotorlapátok elhelyezése után kapcsolj **Move** módba. Ebben a módban az elemeket és a csoportokat tudjuk mozgatni. Ragadd meg a rotortengely felső végét és állítsd egyenesre.

Ha kész a rotor, forgassuk meg. Mivel a tárgy szimmetrikus, elég, ha 90° -ot forgatunk rajta a ciklus során. Ez a negyed fordulat menjen végbe 10 képkocka alatt, a következő kulcskocka tehát a 9. lesz. Bővítsd ki idáig a ciklust majd a **Mode** menüből válaszd ki a **Twist**-et. Ebben a módban a tárgy elemeit a Z tengelyük körül tudjuk forgatni.

Fogd meg a rotortengely tengelypontját az előlnézetben és forgass rajta 90° -ot. Rendeld hozzá minden elemhez (kivéve az összülőt) a csövet, majd az **Animate/Make**-kel készítsd el az animációt. A perspektíva ablakot még az előzetes animáció elkészítése előtt állítsd be, úgy, hogy a legjobb rálátást adja a rotorra.

Ha mident pontosan csináltál, sikerült egy csapnivalóan rossz, ugráló forgást létrehozni. Rövid gondolkodás után rájöhetünk a hiba okára, a 9. kockában ugyan az a rotorlapátok állása, mint az



animációban rákövetkező 0. kockában, ezért a rotor megtorpan, két kockányit tartózkodik ugyan abban a pozícióban. Hogyan küszöbölhetjük ki a hibát? Úgy, hogy a 9. kockában csak 81 fokra forgatjuk el a rotort. Amikor a 10. kockában a ciklust újramezdi, úgy fog tűnni, mintha újabb 9°-ot fordult volna. Elméletileg jó, de Transformation hiányában hogyan állítsuk be a 81°-ot? Bővítsd ki a ciklust a 10. kockáig. Ez automatikusan kulcskocka lesz a 90°-kal elfordított pozícióval. Most lépj vissza a 9. kockára és a **Cell menü UnMake Key** pontjával szüntesd meg a kulcskockaságát. Menj egy kockával arrébb, majd ismét térj vissza ide, ezzel újraszámoltattuk a pozíciót, a rotor 81°-ra van elfordulva a kiindulási pozícióhoz képest. Most a **Cell/Make Key** ponttal tedd kulcskockává ezt a pozíciót. Menj át a 10. kockára és a **Cell/Remove**-val töröld ki a ciklusból. A Remove csak az első, vagy az utolsó kulcskockáját tudja törölni a ciklusnak!

Ha most elkészíted az animációs előzetest, sokkal jobb eredményt kapsz. Ha egy kicsit mégis ugrálna az animáció, az azért van, mert nem pontosan egyformára készítetted a rotorlapátokat, vagy nem pont 90°-ot fordul a tengely. Ezen többféle módon lehet segíteni:

- 1, Precízebben dolgozol.

- 2, 360°-ot forgatsz a tengelyen. Ebben az esetben is figyelj a ciklus loopolhatóságára, vagyis, hogy az első és utolsó kockák jól illeszkedjenek.

A fenti két ajánlást betartva is lehetnek pontatlan-ságok. Szerencsére a cycle objecteket nem csak a Cycle editorban lehet szerkeszteni, hanem a Detailban is. Akkor mire jó a Cycle editor, és miért nem találkoztunk a ciklusépítés funkcióival a Detail ismertetésénél? A Detail editorban csak a ciklus csopotszerkezetét, az egyes elemek helyzetét tudjuk

Cycle editor

beállítani, magát a ciklust mindenképpen a Cycle editorban kell beállítani.

Lépj át a Detail editorba, ahol ismét összeállítjuk a rotort. A csoport őszülőjének hozz létre egy Axis-t. A változatosság kedvéért a rotort ne a megszokott csövünkből állítsd össze, hanem hozz létre új Tube primitíveket.

A rotor alakja a szokásos. Csoportosítsd a tengelyhez a lapátokat, majd a csoportot rendeld alá az Axis-nak. A rotor hierarchiája ezzel kész, de a tegelyek még nem úgy állnak, ahogy a cycle objectben kell. Emlékezz rá, a szülő tengelypontjához kapcsolódik a gyermek Z tengelyének vége. A tegelyek beállítását segíti a Detail editorban a Functions menü **Cycle Setup** pontja. Ez úgy állítja be a tengelyeket, hogy azok megfelejenek a Cycle objectnek.

A funkció nem tudja (mert nem tudhatja) beállítani a forgáspontok helyét, ezt nekünk kell megtenni az interaktív transzformációkkal, vagy Transformation-nal. a Transform Axes Only-t bekapcsolva. Ne feledd, az interaktív transzformációk billentyűit a shift-tel együtt használva a művelet csak a tárgy tengelyén hajtódik végre.

Ha kész a forgáspontok beállítása, Pick Group-ban válaszd ki a teljes tárgyat és használd a Cycle Setup-ot. Ezután mentsd ki az egészet rotor-0.iob néven.

Forgasd el a csoportot a Z tengelye körül 81° -kal. A pontos forgatást természetesen a Transformation-nal lehet megtenni. Az elforgatott csoportot mentsd ki rotor-9.iob néven.

Lépj át a Cycle editorba és a Load-dal töltsd be a 0. kockába a rotor-0.iob-ot. Egy kérdező jelenik meg, mely tájékoztat, hogy tárgy nem cycle object, átkonvertálja-e. Válaszolj Yes-szel.

Rövid konvertálás után megjelenik a rotort alkotó

csopot struktúrális és perspektivikus képe. Most bővítsd ki a ciklust a 9. kockáig. Ebben még az alap állású rotor van. A **Cell menü Load Pose** pontjával töltsd be a rotor-9.iob tárgyat, amely most a már meglévő csoport új pozícióját határozza meg. Készítsd el az előzetes animációt. Ebben már sokkal szebb a rotor forgása.

A Load Pose használatára van néhány megkötés. A pozíciót meghatározó csoport hierarchikus szerkezetének azonosnak kell lenni a cycle object szerkezetével. A pose group-ot alkotó tárgyaknak nem kell azonosnak lenni a cycle object tárgyaival, a Load Pose csak a tárgyak tengelyeinek pozícióját veszi át.

A pozíciót adó tárgyaknak meg kell felelni a ciklikus tárgyak elvárásainak (tengelyek elrendezése).

Pozíciót csak kulcskockában lehet betölteni.

A Load Pose segítségével bonyolult cycle objecteket is pontosan tudunk felépíteni, az Object/Load-dal Detail editorban létrehozott csopotot betöltve a ciklikus tárgy precízen építhető fel.

Időnként szükség lehet arra, hogy a ciklikus tárgy valamely fázisát pontosan beállítsuk, annak ellenére, hogy a csoport nem a Detail editorban lett összeállítva. Erre a célra használható a **Cell/Snapshot**. A funkció az adott kockában lévő csoportot (nem szükségszerűen kulcskocka) kimentti normál csoportként, amelyet azután a Detail-ban továbbalakíthatunk, vagy Load Pose-zal bárhova (csak kulcskockába) betölthetünk.

A Cycle editorral való ismerkedésünk utolsó fázisaként készítsük el az alábbi robotkart. Ami lényeges benne, hogy az ujjai azonos hierarchikus magasságon vannak, mégsem egy pontba csatlakoznak, forgáspontjuk között van egy bizonyos távolság.

Hogyan valósítható ez meg? A null link használata

Cycle editor

tával. A null link egy olyan csoportelem, amelyhez nem rendelünk semmilyen tárgyat. A null link fenntartja a csoportkapcsolatot, de a képen nem látható.

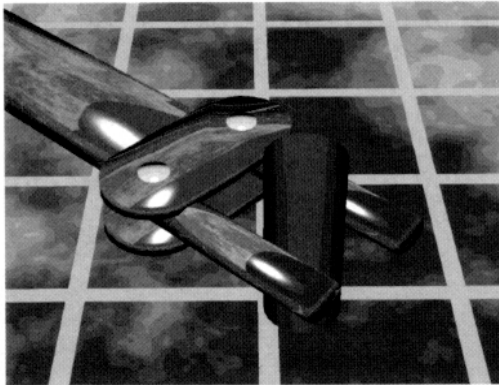
Az említett robotkarnál a robot kézfejehez két null link kapcsolódik, egy mástól széttartva, az ujjak pedig ezekhez a null linkekhez kapcsolódnak.

Null linket a Cycle editorban úgy lehet létrehozni, hogy a null linknek szánt rombuszhoz nem rendelünk tárgyat. A Detail editorban a null link egy megfelelően beállított Axis.

Mielőtt rátérnénk a ciklikus tárgyak felhasználásának ismertetésére essen szó a Detail editor Functions menüjének két, a cycle object-ekkel kapcsolatos pontjáról.

Amikor egy cycle object-et a Detail-ban manipulálunk, előfordulhat, hogy a tengelyek helyes láncolata megszakadt. Például a csoport egy elemét elmozgatva, vagy elforgatva a tengelypont-Z tengelyvég kapcsolat felbomlik. Ennek korrigálására szolgál a **Cycle Shuffle** menüpont. A csoporton végrehajtott változtatás után, Pick Groups módban ki kell választani a korrigálandó elemet, majd használni a Cycle Shuffle-t. A funkció helyreállítja a cycle object-nek megfelelő tengelyszerkezetet. A korrekció közben legtöbbször a csoport alakja is megváltozik.

A másik ciklikus tárgyakkal kapcsolatos opció, a **Cycle Transform**. Ez egy kapcsoló, kipipálása után a tárgyakon az interaktív forgatás nem a tengely-



pont, hanem a Z tengely vége körül történik, megfelelően a ciklikus tárgy mozgásának. A kapcsoló csak interaktív forgatásnál fejt ki a hatását, a Transformation továbbra is a tengelypont körül forgat.

Hogyan állíthatjuk be a cycle object attribútumait? Mint láttuk, a ciklikus tárgy normál Imagine objektumból áll, amelyek attribútumainak beállítását ismerjük. A kész cycle object is betölthető a Detail editorba, ahol a szokott módon állíthatjuk be az azt alkotó tárgyak tulajdonságait.

A ciklikus tárgy felhasználását a Stage és Action editorokban az ingákon fogjuk gyakorolni. Nyiss egy új project-et és lépj be az Action editorba.

Készíts egy 40 kockás animációt. Tedd fel a színre mind a 40 kockában az ingát a szokott módon, normál tárgyként. A megjelenő **Object File Info** kérdezőben van két input mező és egy kapcsoló, amely a ciklikus tárgyak kezelésével kapcsolatos, eddig még nem esett róluk szó.

A **Number of cycle to perform** mezőben kell megadni, hogy a cycle object hányszor ismétlje el a kijelölt kockák alatt a mozgásciklusát. Ez az érték lehet tört szám is, ekkor az utolsó ciklus nem zajlik le teljes egészében. Ha ide nullát írsz, a ciklikus tárgy nem nem hajtja végre a mozgássorozatát.

Például az ingákhoz 1.5-öt írva az utolsó kockában az inga nem a bal, hanem a jobb oldali végpozícióban lenne. Tegyen most az inga két egész lengést.

Az **Initial cycle phase (0.0....1.0)** egy arányszám, annak az aránya, ahonnet a teljes ciklushoz képest a tárgy indul. Például az ingánkat a 0.5 értékekkel indítva az első kockában a ciklus középső fázisa fog megjelenni, vagyis a jobb oldali végállás. A 0 érték

Cycle editor

jelenti, hogy a ciklus az elejéről indul, az 1, hogy a végétől.

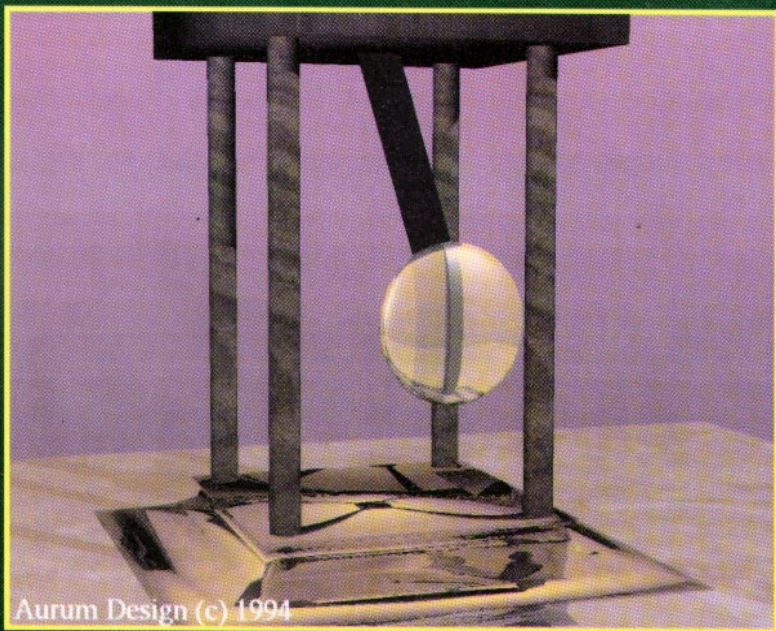
A Reverse cycle motion kapcsolót kiikszelve a tárgy mozgása visszafelé történik, a ciklikus fordítva megy végbe.

Ennyi az összes tudnivaló, amely a cycle object felhasználásához szükséges.

A Stage editorban a ciklikus tárgy úgy kezelhető mint egy normál object, pozícióját meghatározni az ősszülő tengelyén keresztül lehet, csakúgy mint a normál csoportokét.

Állítsd be a színpadot ahogy gondold és készítsd el az animációs előzetest. Az inga a várakozásnak megfelelően kétszer hajtja végre a lengést.

Próbáld ki az animációt többféle perioduszammal és figyelj meg, hogyan igazítja hozzá a program a ciklus időzítését a körülményekhez.



Forms editor

A Forms editor - méltatlanul - a legkedvesebbet használt szerkesztője az Imagine-nek. Sokan, akik nem ismerik és nem tudják használni, fölöslegesnek, használhatatlannak tartják. Most szeretném ezt a tévedést kiigazítani és megmutatni, hogy a Forms editor milyen jól használható bonyolult felépítésű szimmetrikus vagy asszimmetrikus tárgyak alakjának elkészítéséhez.

Bár a forms object önnálóan is felhasználható a Cycle, Action és Stage editorokban, mégis legtöbbször a Detailban tovább szoktunk alakítani, hiszen például az attribútumokat csak ott tudjuk beállítani.

A forms object struktúrája némiképp eltér a normál objectekétől, a Forms editorba csak a forms objectet lehet betölteni, amit a Detail-ból mentettünk ki, vagy a Stage és Cycle editorokból a Snapshot funkcióval exportáltunk, nem.

A Forms editor nagy erőssége a szimmetrikus tárgyak létrehozása. A tárgyakat metszeteken keresztül tervezhetjük meg. Ideális lehet például hajótest, repülőgép-szárny, gépkocsi karosszéria, stb esetében. Ennek demonstrálására töltsd be a program demo lemezéről a **Car.forms** tárgyat. Ez egy autó körvonalát mintázó forms object. Elkészítése a forms editorban kb. 15-20 perc (kellő gyakorlat birtokában). Ugyan ezt a tárgyat a Detail-ban elkészíteni több órába kerülne.

A Forms editor **Project** és **Display** menüi teljesen szokványosak.

Az **Objects** menüben a **New**, **Load** és **Save** mellett találjuk a **Snap to Grid** pontot is, amellyel a kiválasztott pontokat a hozzájuk legközelebb eső rácspontokra igazíthatjuk. Ezt a funkciót már ismerjük a detail editorból.

Forms editor

Az ismerős menüpontokhoz tartoznak a **Select** menü elemei, amelyek a pontkiválasztás metódusát szabályozzák. **Click**-ben a pontra kell klikkelni, **Drag Box**-ban egy gumikerettel, míg **Lasso**-ban egy szabadkézi vonallal kell bekeríteni a kiválasztani szándékozott pontokat. **Multi select** a shift használatával lehetséges.

A **Lock** bekapcsolása után a manipulált pontok mindig a legközelebbi rácspontra igazítódnak.

A **Mode**, **Symmetry** és **Cross Section** menük pontjainak funkcióit e fejezet során fogjuk részletezni.

Mielőtt belemélyednénk a részletekbe, meg kell ismerkednünk a forms objectek felépítését. Néhány sorral korábban említettem, hogy a Forms editorban a tárgyakat metszeteken, szeleteken keresztül tudjuk megtervezni. Maga az eljárás hasonló, mint mikor a Detail editorban elkészítettük a hajó bordázatát és a Skin funkcióval felületet húztuk rá.

A forms objecten tetszőleges számú metszetet vehetünk fel. Fontos, hogy minden metszet azonos számú pontból áll. Minden szelet egymásnak megfelelő pontjait összekötve jön létre a tárgy felülete. A művelet automatikus, ezzel a felhasználónak nem kell törődni. A metszetek alakja szabadon változtatható.

Amikor valamely metszeten új pontot helyezünk el, vagy egy meglévőt törölünk, minden metszetben megjelenik vagy törlődik egy, a manipulálnak megfelelő pont.

Miért előnyös a Forms editor használata?

A tárgy metszetei jól átláthatóak, könnyen és gyorsan lehet kialakítani még az összetett formákat is. Az elkészült tárgyak felülete mindenütt egyformán összefüggő. A Detail editorban primitívekből összerakott, extrudált tárgyak sokszor tartalmazznak

Forms object készítése

töréspontokat, feltűnően illeszkedő éleket.

Külön funkciók segítik a pontos szimmetria kialakítását. Detail-ban ez nagyon nehézkes.

Mivel a tárgyak geometriája igen kötött, könnyen készíthetők morfózisra alkalmas tárgyak.

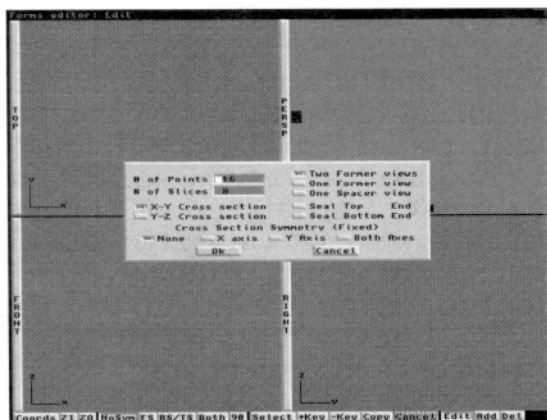
Az előnyök mellett vannak azonban hátrányok is, a Forms editor semmiképp nem teszi feleslegessé a Detail-t, de azt nagyon jól kiegészíti. A tárgy részleteinek kidolgozására, tulajdonságainak beállítására csak a Detail editorban van lehetőség.

Forms object készítése

A Forms object készítését az **Object** menü **New** pontjának kiválasztásával kell kezdeni. Ezzel létrehozunk egy kiindulási tárgyat, amelyből az egyes metszetek megformálásával kialakítjuk a tárgyat. Lényeges különbség a Detail és Forms editorok között, hogy utóbbiban mindig egy alap tárgyat alakítunk.

A New használata után megjelenik egy kérdező, ahol a kiindulási tárgyra és a szerkesztés mikéntjére vonatkozó dolgokat állíthatjuk be.

A **# of Points** a metszetet alkotó pontok száma, a **# of Slices** pedig a metszetek száma. Ha később, a szerkesztés folyamán kiderül, hogy kevés, vagy ellenkezőleg, túl



Forms editor

sok a szeletek vagy a pontok száma, növelhetjük vagy csökkenthetjük azokat.

A **X-Y Cross section** és **Y-Z Cross section** kapcsolókkal lehet kiválasztani, hogy a metsző síkok hogyan helyezkedjenek el. Az X-Y síkkal álló tárgyakat, mint például a vártorony, az Y-Z síkokkal pedig fekvő tárgyakat, mint a hajótest, repülőgépszárny, stb készíthetünk. A metszősík helyzete a tárgy alakját nem befolyásolja, csak a szerkesztő ablakokban való elhelyezkedését.

A **Two Former views**, **One Former view** és **One Space view** kapcsolókkal a megjelenítés módját lehet beállítani, erről majd később esik szó.

A **Seal Top End** és **Seal Bottom End** kapcsolókkal a tárgy felső és alsó végét tudjuk zártra készíteni. Ha az Y-Z Cross section-t választottuk, akkor a tárgy fekvő helyzetű, ezért a kapcsolók a jobb és bal oldali végeket zárják be.

A **Cross Section Symmetry (Fixed)** kapcsolóival a metszetek szimmetrikusságát lehet szabályozni. A **None**-t választva a metszetet alkotó minden pont önállóan, a többitől függetlenül manipulálható. Az **X** vagy **Y** bekapcsolása után kiválasztott tengely egyik oldalán elmozgatott ponttal, vagy pontokkal együtt mozdul a tengely túloldalán lévő szimmetriapárja is. Ahhoz, hogy e két szimmetria valamelyikét használni tudjuk, a metszeteknek páros számú pontból kell állni. Ha az Y-Z Cross section-t kapcsoljuk be, a tárgy fekvő helyzetű lesz, a szimmetria az Y és Z tengelyek viszonylatában valósul meg.

A **Both Axes**-t választva a szimmetria egyszerre érvényesül mindkét tengelyre, azaz egyszerre négy pont fog szimmetrikusan mozdulni. Ebből következik, hogy a metszet négyvel osztható számú pontokból kell hogy álljon.

Forms object készítése

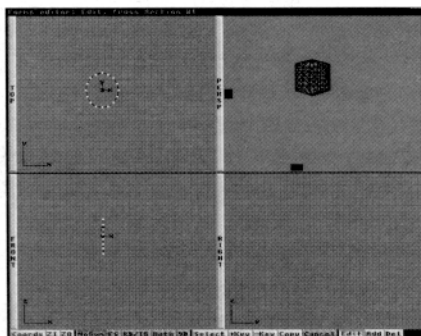
Első forms objectünk legyen egy asztalláb, amely hengeres és négyzetes elemekből áll. A metszetet alkotó pontok száma legyen 16, a metszetek száma pedig 8. Az asztalláb álló formátumú tárgy, ezért X-Y metszősíkokat alkalmazunk az elkészítéséhez. A megjelenítés módja legyen One Spacer view. Az asztalláb mindkét vége zárt, kapcsoljuk be ezeket az opciókat. A metszetek legyenek mindkét tengely mentén szimmetrikusak. (Both Axes).

Ha a beállítások készek, klikkelj az OK-ra. A megjelenő kiindulási tárgyat megvizsgálva tapasztalhatjuk, hogy a szerkesztő ablakokban lévő kép nem a szokványos. A felülnézet helyén az éppen szerkesztés alatt álló metszet körvonala látható, a Front nézetben egy egyenes mentén helyezkednek el a metszősíkok pozícióját jelentő pontok, míg a jobbnézet üres. Ez az elrendezés az One Spacer view-re jellemző.

Ha az Y-Z Cross section-t választjuk, a tárgy más orientációjából adódóan a felül nézet helyett a jobb nézetben lesz a metszet körvonala, a front nézet funkciója változatlan.

A perspektíva ablakban a tárgy pontos perspektívus képe látható. A forms object valódi alakját csak it tekinthetjük meg.

A front nézetben látható, hogy a metszősíkok egymástól egyforma távolságra helyezkednek el. A pontokat vonalak kötik össze, ezek jelölik, hogy amikor a metszetre felületet húz a program, az egyes metszetek milyen sorrendben kötődnek össze. Most a



Forms object készítése

lősíkokat jelképező pontokat és azok széthúzóztatásával nyújtsd meg a tárgyat. Ügyelj arra, hogy a pontok eredeti sorrendje és a henger csúcsossága is megmaradjon. A metszősíkok eloszlása legyen olyan mint a képen. Ha a rácssűrűséget kisebbre állítod, könnyebb lesz a feladatot megvalósítani.

Térj vissza a negyed nézetbe és a rácssűrűséget is állítsd az alap 20-as értékre. A metszetek szerkesztéséhez ez a Zoom faktor nem megfelelő, de még hagyjuk így.

Mivel a felülnézetben egyszerre csak egy metszet képe látható és szerkeszthető, valahogy ki kell választani melyik legyen az . Erre a **Cross Section menü Select** pontja, vagy az azonos felíratú User Gadget szolgál. Próbáld is ki mindjárt. Az eredmény egy hibaüzenet, mely szerint a jelenlegi metszet az egyetlen, amely definiálva van.

Ennek megértéséhez tudnunk kell, hogy kétféle metszettípus van: kulcsmetszet és közbeszúrt. A kulcsmetszet azt jelenti, hogy annak alakját a felhasználó határozza meg, míg a közbeszúrt metszeteket a program automatikusan készíti el a kulcsmetszetek közé szűrve, azok közötti átmenetként. Kezdetben egyetlen kulcsmetszet van, a legfelső, Y-Z metszősíkoknál a jobb oldali.

Ahhoz, hogy az asztallábat megformálhassuk ki kell jelölni a definiált metszeteket. Válaszd ki a **Cross Section menü Make Key** pontját. Az állapot-sorban megjelenik a "Válassz egy metszete, definiálni azt kulcsként". A már kulcsként definiált metszetek színe narancs lesz.

Klikkelj arra a pontra, amelyhez tartozó metszete kulcsként akarod definiálni. Készíts minden metszeteből kulcsmetszete.

Mielőtt tovább mennénk, elmondom, hogy a kulcsmetszeteiket az **Unmake Key** menüponttal

Forms editor

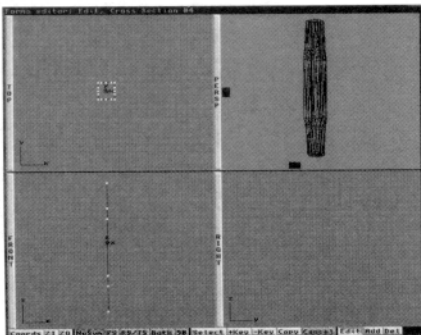
lehet ismét közbeszúrt metszetté alakítani. Az eljárás azonos a Make Key-jel.

Miután minden metszetet kulcsmetszetté tettük, kezdhethetjük az asztalláb kialakítását. Mindössze annyit kell tennünk, hogy a középső két metszetet négyzet alakúra formázzuk. Válaszd ki valamelyiket, a sorszáma megjelenik az állapot sorban a Cross Section # után. (Megjegyzem, amikor egy metszetet kulccsá teszünk, automatikusan az lesz a szerkesztett, kiválasztott metszet.)

Állítsd a Zoom faktort 1-re, majd a felülnézetet nagyítsd teljes képernyőre. Mozgasd úgy a pontokat, hogy a metszet képe egy négyzetet formázzon. A kettős szimmetria és a Lock segítségével ez néhány pillanat alatt megvan. Jegyezd meg, hogy ez melyik metszet volt, majd válaszd ki a másikat. Ezt még gyorsabban négyzetesre alakítjuk, válaszd ki a **Cross Section** menü **Copy From** pontját, majd klikkelj arra a pontra a front nézetben, amely az előző szeletet jelképezi. Ezzel átvettük annak az alakját.

Az asztalláb majdnem kész, már csak annyi a teendőnk, hogy a két végéről eltüntessük a csúcsokat. A front nézetben a nekik megfelelő metszősíkot a szomszédos síkkal azonos pozícióba kell vinni. Azért nem tettük ezt idáig meg, hogy ne okozzon zavart az egymást fedő, azonos pozícióban lévő két metszősík.

Próbáld meg ismét elkészíteni az előző tárgyat, most már önállóan, teljes folyamatában. Ugye egyszerűbb, mint a Detail-ban primitívekből összerakosgatni, kinyomni a síkot, felületeket



Forms object készítése

kreálni, stb? Pedig ez a lehetőségeknek csak piciny része.

Van a Cross Section menünek egy ötödik pontja, amiről még nem esett szó, de gondolom mindenki rájött, mire való. A **Cancel**-lel a Cross Section menü másik négy funkciójából léphetünk ki a végrehajtásuk nélkül.

Folytassuk a Forms editorral való ismerkedésünket, készítsük el egy gépjármű motorjának főtengelyét. A tárgy felépítéséhez használunk most 12 metszetet, egyenként 16 ponttal.

A főtengely általában fekvő helyzetű, ezért használunk Y-Z metszősíkokat. A metszetek alakján nem fogunk változtatni, ezért a metszeti szimmetria lehet None.

Megjelenítési módul válasszuk a One Former view-et és tegyük mindkét végén zárttá a tárgyat. OK. A megjelenő "**Keep the old forms?**" kérdésőben válaszoljunk No-val. A "**Keep the old cross section points?**"-ra szintén.

Ezek magyarázatát később adom meg.

A létrejövő kiindulási forma egy gömb. A metszősíkok pozícióját most is a front nézetben látjuk, de minden síkhoz két pont tartozik. One Former view, megjelenítési módban az egyes metszeteket nagyíthatjuk és elbillenthetjük, azaz itt már nem föltétlen párhuzamosak a metszetek. A jobb nézetben lehet a kiválasztott metszetet szerkeszteni. Indulásként most is egy kulcsmetszet van, a jobb szélső.

A metszeteket összekötő vonalak ebben a módban is a felületképzés sorrendjét jelölik.

One Former View módban már van értelme olyan tárgyat létrehozni, ahol a pontok sorrendje nem azonos az összekötésük sorrendjével. Ilyen tárgy például az esernyő, ahol a nyélen lévő magasabb

Forms editor

pontból az esernyő vászna visszahajlik egy alacsonyabb pozícióba.

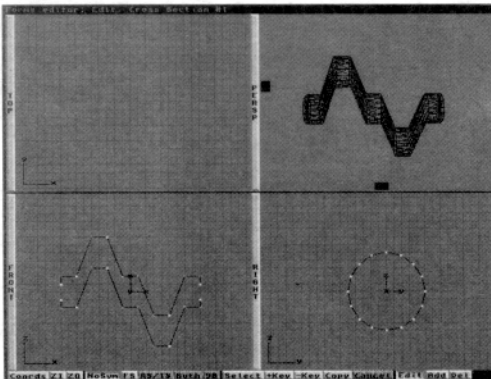
A főtengely számára nem a gömb a legideálisabb forma, ezért át kell alakítani azt egy csővé. A metszeteiket két pont jelképezi, amely pont a metszet átellenes végein helyezkedik el. A pontokat egymáshoz közelítve a metszet mérete csökken, míg távolítva növekszik. Az egyik pontot hosszirányban elmozgatva a metszet elbillenthető, mindkét pontot együtt mozgatva a metszet a középvonalból kivehető. Bármelyik műveletet is alkalmazzuk a metszet alakja nem változik.

Az One Former view-ban a metszetek két tengely mentén mozgathatók (a hossz és az egyik kereszt tengely), méretváltozásuk mindkét kereszt-tengely mentén azonos mértékű.

Ebben a megjelenítési módban választhatunk szimmetriát a metszet manipulálásához. Ez nem azonos a New kérdezőjében a metszet alakjára beállítottal, hanem a front nézetben a metszősíkon végrehajtott műveletre vonatkozik. Két opció közül választhatunk a Symmetry menüből. Az Off-ot kipipálva nincs szimmetria, míg a Frony View-et használva, ha a metszetet jelképező egyik pontot

mozgatva, a párja vele szimmetrikusan fog mozdulni. Próbáld ki kicsit a metszősíkok manipulálásával, figyeld az eredményt a perspektíva ablakban.

Kapcsold be a szimmetriát és a Lock-ot, majd készíts el egy keskeny,



Forms object készítése

hosszú csövet kiindulási alapként. A csúcsok eltüntetését most is hagyjuk a végére.

A metszetek alakját nem fogjuk megváltoztatni, ezért nincs is szükség kulcsmetszetekre. Nagyítsd ki a frontnézetet és alakítsd ki az egyes metszetek elhúzogatóásával a képen látható formát. Hogy a pontok együtt, egy irányba mozduljanak, a szimmetriát ki kell kapcsolni és multi select-tel kell a páros pontokat elhúzni.

Gyakorlatilag az utolsó néhány művelet volt az, amellyel a főtengelyt kialakítottuk. Mentsd ki, majd a Detai-ban lásd el attribútumokkal és rendereld le.

Próbáljunk most egy olyan, az előzőhöz hasonló tárgyat készíteni, amelyen nem csak Z, hanem Y irányban is van göbület. One Former view-ban ezt nem tudjuk elérni, az új kiindulási tárgyhoz válassz **Two Former view**-t. A metszősík továbbra is Y-Z.

A megjelenő kiindulási tárgyunknak a felülnézetben lesz egy másik metszetsorrendi képe. Minden metszethez négy pont tartozik, a front nézetben, lévő két pont a metszősík 0° és 180° -nál lévő pontjai, míg a felülnézetben lévők a 90° és 270° -nál elhelyezkedő pontok.

Two Formers view-ban ötféle metszősík szimmetria opció van.

None - nincs szimmetria, a metszősíkot jelképező minden pont a másiktól függetlenül manipulálható.

Front view - Az előlnézetben lévő két pont, a 0° és 180° pontjai egymással szimmetrikusak, függetlenül a másik két ponttól.

Right/Top view - a jobb nézet (Y-Z metszősík esetén a felülnézet) pontjai, a 90° és 270° -nál lévők egymással szimmetrikusak, függetlenül a másik két ponttól.

Forms editor

Both - Az előző kettő kombinációja, mintha mindkét nézet szimmetriája be volna kapcsolva. 0° és 180° pontjai egymással és a 90° és 270° pontjai is egymással szimmetrikusak.

90° Degree - Középpontos szimmetria, amely a metszetet jelölő mind a négy pontra érvényes. Gyakorlatilag mind a négy pont együtt, szimmetrikusan mozdul.

A Two Former view, amellettt hogy lehetővé teszi a metszősíkok minden irányú mozgását, másra is alkalmas. Mind a négy pontot (nem szimmetrikusan) oldalra húzva a metszet eltávolítható a középvonalból. Gyakorlatilag ez olyan, mint az előbb a főtengely kialakítása, csak mindkét kereszt tengely mentén lehetséges. (A négy pontot csak kettessel, nézetenként külön-külön tudjuk mozgatni.)

Valamely nézet két pontjának széthúzásával-összenyomásával a metszet oválissá válik. Ha 90° Degree szimmetriát választunk, a metszet mindkét tengely mentén azonos mértékben változtatja méretét, azaz nem lesz ovális!

A metszetet jelölő két pontot hosszirányba elmozgatva a metszet meggörbíthető, mint például egy lovaglő nyereg, vagy a cylinder karimája. Ha csak a metszetet jelképező egyik pontot mozgatjuk el, a metszet széle "felhajlik". Ezt úgy képzelj el, mintha a metszet egy papírra lenne rajzolva és a lap egyik sarkát megemelnéd (vagy másik irányú elmozgatás esetén lenyomnád a többi alá.)

A metszősík semmilyen transzformációja nem befolyásolja magának a metszetnek az alakját a szerkesztő, ablakban. Hogy miért nem? Mert az eltorzított metszősíkról, amin a metszet alakja ténylegesen megváltozott, a sík, derékszögű szerkesztő ablakba vetítjük vissza. A metszet alakja vetítéskor "visszatorzul". Az előbb leírtak, főleg

Forms object készítése

kezdők számára, belátom kissé zavarosak, de mindenki gyakorolgassa egy kicsit, figyelje meg, mi mivel van összefüggésben és a metszetek manipulációja hogyan változtatja meg a tárgy drótvázának szerkezetét. Rövid ismerkedés után a Forms editor is könnyen kezelhető, hatékony eszközzé válik.

A tárgyak kialakításához nem mindig elegendő, vagy épp túl sok az eredetileg meghatározott pontok és metszetek száma. Szerencsére bármikor adhatunk vagy törölhetünk a metszet körvonalát alkotó pontot, vagy teljes metszetet.

Új elem hozzáadását az **Add** mód bekapcsolása után lehet végezni. Valamely metszetet jelölő pontra klikkelve, azt a pontot az egyik szomszédjával összekötő egyenes közepén új pont, azaz metszősík jön létre. A metszet körvonalán ugyan így hozhatunk létre új pontot, de ez a pont a neki megfelelő helyen, minden metszeten létrejön.

Metszetet vagy pontot törölni hasonlóan lehet, ehhez **Delete** módban kell lenni. A megfelelő pontra klikkelve az vagy a hozzá tartozó metszősík, törlődik. A metszet körvonalát alkotó pont törlésekor óvatosan kell eljárni, mert az adott pont minden metszetről törlődik. Ha nem vagy biztos, hogy a pont mindegyik metszeten felesleges-e, ne törtöld a pontot, inkább illeszd be a körvonalba. A néhány szükségtelen pontot tartalmazó tárgy nagyobb mérete kisebb baj, mint az akaratlanul törölt pont.

A tárgy bizonyos paramétereit, a megjelenítés módját, stb, csak a kiindulási tárgy létrehozásakor, a New kérdésjében lehet beállítani. Mit tegyünk például olyan esetben, amikor egy félig elkészített tárgynál kiderül, kevés a One Former view által nyújtott szerkesztési lehetőség, szükség volna a Two Formers view-ra? Eddigi ismereteink szerint újra kell

Forms editor

kezdeni a szerkesztést ebben a megjelenítési módban. Következően jobban meg kell választani a kezdeti paramétereket.

És mit tegyünk, ha a metszetek elkészítésekor az egyébként szimmetrikus alakok közt asszimmetrikus elemek is vannak? A metszet szimmetriáját szintén a New kérdésében lehet beállítani. Nem csigázlak tovább, elárulom, ezek az alapvető paraméterek is megváltoztathatók. Hozz létre egy tárgyat tetszés szerinti paraméterekkel. A látvány kedvéért adj hozzá egy új metszetet és töröld le egy pontot a körvonalról. Szerkessz egy kicsit az alakján is.

Változtassuk most meg a metszetszimmetriát. Válaszd ki az Object/New-t. A biztonsági kérdésre, a "**Last saved unchanged, proced?**"-re válaszolj Yes-szel. Megjelenik a szokott kérdéső. Figyeld meg a pontok és szeletek számát az input mezőkben! Azok tükrözik a tárgyon tett változtatásokat, az aktuális állapotot mutatják. Allítsd át a szimmetria kapcsolóját majd klikkelj az OK-ra.

Megjelenik egy kérdéső, "**Keep the old Forms?**" felirattal. Válaszolj Yes-szel, minek hatására az új tárgy szeletelő síkjainak pozíciója azonos lesz a régivel.

Újabb kérdéső, "**Keep the old cross section points?**". A Yes itt azt jelenti, hogy a szeletek alakja azonos lesz az előző tárggyal.

A két Yes-szel az új tárgyban a metszetek elhelyezkedése és azok alakja is változatlan marad, ezért a két tárgy azonos lesz. Az egyéb változtatások, mint a szimmetria, megjelenítési mód, végek zártsága, srb, megváltozik, gyakorlatilag tehát csak ezek változnak, ami pontosan megfelel az eredeti célunknak.

A szeletek elhelyezkedését és alakját csak akkor lehet megtartani, ha a pontok vagy metszetek száma nem változik. Például ha a New kérdésében a

Forms object készítése

szeletek számát megváltoztatjuk, csak a szeletek alakját tudjuk megtartani, azok elhelyezkedése alapértelmezés szerint lesz.

Ugy érzem mindent elmondtam a Forms editorról, amit el akartam. Ahhoz, hogy hatékonyan tudd használni, már csak gyakorolni kell, de ez igaz a többi szerkesztőre is.

A Forms editor leírásának befejezésével a finisbe értünk, megismertük a tárgyszerkesztést, kép és animáció készítés minden funkcióját. Ami még hátra van, az nem kapcsolódik szorosan az előbbi témakörhöz.

Ahhoz, hogy szép képeket és animációkat készíts, már csak olyan dolgok hiányoznak, amelyek nem adhatók át a könyvön keresztül: ötlet, kitartás, kísérletezés, gyakorlat. Innen kezdve minden rajtad múlik. Persze adódhatnak olyan helyzetek, amelyeket nem tudsz egyedül megoldani, nem jössz rá a megoldásra, ebben segít az Aurum DTP Stúdió Imagine tanácsadó szolgálata. Fogalmazd meg a problémát érthetően, írd meg levélben a stúdió címére, mellékelj egy válaszborítékot, ha a kérdés egy képhez kapcsolódik, akkor egy lemezt a képpel, object-tel, stage fájljal, stb. A hozzánk érkezett kérdésekre válaszolunk (ha tudunk), a problémák megoldására tanácsot adunk. Mindez a könyv olvasóinak teljesen ingyenes.

A kiadó szívesen fogad képeket, animációkat, ötleteket is. Írjátok meg, egy-egy jól sikerült képen milyen új trükköket alkalmaztatok. Az Imagine könyv következő kötetébe a legszínvonalasabb alkotások, ötletek bekerülnek, szerzőjüknél honoráriumot fizetünk.

Mi mást kívánhatnék befejezésképpen? Kellemes ray-tracinget. Nemsokára ismét találkozunk.



A zongoránál maga Beethoven ül. A fej az Imagine demóobjectje, a test és a zongora saját találmány. A talaj egy kockás textúrával ellátott ground. Ez a tárgy végtelen kiterjedésű, látszik is a mintázatán, amely erősen torzul a távoli részen. Ilyen esetekben célszerűbb inkább plane primitívet használni, azon nem lép fel ez a hiba. A ground jelentősége abban



áll, hogy már scanline képszámítási módban is látszik a tükröződő felületekben.

A bal oldali képen a hullámzó vizet a Ripple effekttel hoztam létre. Ehhez tíz kocka hosszúra állítottam az objectet, de csak a középső képet generáltam le.

Reset - Minden beállítást a gyári alapértékre hoz.

Save - Elmenti a Preferences beállításait az Imagine tartalomjegyzékébe, Imagine.config néven. (PC-n Imagine.con)

Last Saved - Betölti az Imagine.config fájlt, felülírva ezzel a jelenlegi beállításokat.

Save As - Kimenteti a beállításokat a felhasználó által adott néven.

Load From - Betölt egy nem alap néven kimentett konfigurációt.

Miután tisztába jöttünk a körítéssel, ismerkedjünk meg a lehetséges beállításokkal is. Kezdjük a legelején, a Misc.Stuff-fal. Ez tartalmazza az általános dolgokat, amik a program működését befolyásolják. Ilyenkor nincs jelentősége annak, hogy melyik editor gadgetet kapcsoltuk be. A képernyőn megjelenik egy hosszú lista, amely a következő elemekből áll:

Mnemonic - Négybetűs emlékeztető szócska, arra utal, hogy abban a sorban mit lehet beállítani.

Type - A paraméter típusa. Ez lehet:

Text, ekkor valamilyen szöveg a paraméter

T/F, ha igaz-hamis beállítás között kell dönteni

RGB, a paraméter egy szín, összetevőként 4 biten megadva

RGB8, a paraméter itt is egy szín, de összetevőként 8 biten, a teljes szín tehát 24 bites.

Integer, valamilyen egész szám

Float, a paraméter egy lebegőpontos érték

Value - A paraméter

Comment - Magyarázó szöveg.

Az egyes értékeket úgy változtatjuk meg, hogy a sorukba klikkelünk. Az a sor világosabb háttérű lesz, a Value és a Comment paraméterei megjelennek az alsó két input mezőben, ahol meg lehet változtatni azokat. A változtatást mindkét sorban enterrel kell

Preferences

lezárni, ekkor kerülnek be a helyükre. Ha RGB értéket kell megadnunk, a jobb alsó három tolóka és a színellenőrző négyzet a segítségünkre van, de beírni nekünk kell az értékeket.

Az egyes Mnemonic-ok jelentése a következő:

PPTH - Picture Path. Az Imagine főcímkép elérési útvonala. Mivel a kép általában abban a könyvtárban van, mint a program, amely könyvtár a Default Directory lesz az Imagine elindítása után, itt "" az alapérték.

EDIT - Editor. Annak a szövegszerkesztőnek a neve, amelyet a program az animációs lista szerkesztésére kínál fel.

QUIK - Quickrender Style. Az a képernyő formátum, amit az Imagine a Quickrender módhoz használ, pl. Lores Quarterscreen. Ennek a stílusnak szerepelnie kell a preset listában.

QURM - Quickrender Render Method. Itt adhatjuk meg a Quickrender eljárását. Azok közül választhatunk, amit a Project editor Modify kérdésében a Rendering Method-nál találunk.

QUFF - Quickrender File Format. A gyors eljárással készített kép fájlformátuma. Azok közül választhatunk, amit a Modify kérdésben felkínál a program.

QPTH - Path for QR. Picture. A QR. kép elérési útvonala. Ide menti ki a program a képet, amit ha nem töröltetünk le, vagy nem generálunk újat, később is felhasználhatunk. A kép neve Quickrender.

LOAD - Load all modules? A program képes overlay módban is működni, amivel jelentős memóriamegtakarítás érhető el. Ilyen esetben csak az aktuális editor programrésze van a memóriában, a másikat átlépéskor tölti be. Ha paraméter T azaz true, a teljes program egyszerre a memóriában lesz, ha F, false, overlay üzemmódban működik.

Preferences

LACE - Interlace editors? Ha a paraméter T, az editorok interlace-esek lesznek, azaz a függőleges felbontás kétszeresére növekszik. Nem multisync monitoron ez a felbontás remegni fog.

GRON - Grid lines on? Ha a paraméter T, a szerkesztőkben a segédrács be lesz kapcsolva, ha F, akkor nem. A Grid On/Off menüponttal ezt felül lehet bírálni.

BWLN - Make line in B&W shade? B/W shade render módban a tárgyakat határoló háromszögek körvonalait megjelenítse-e.

COLN - Make line in Color Shade? Mint az előző, de Color Shade módban.

WARN - Warn about unsaved changes? Figyelmeztessen-e ha nincs kimentve egy változtatás, pl. az Action editor elhagyásakor.

WRNO - Figyelmeztessen- e már meglévő object fájl felülírásakor.

SREV - Reverse lenses in 3d stereo. Megcserélje-e 3D Stereo módban a lencsék helyét. Balszemesek előnyben.

GNEW - Inital "Generate New Cells Only" button state. A kapcsoló alapállapota.

BGRD - Background color. A szerkesztők háttérszíne.

FGRD - Foreground color. A vonalak színe.

REQC - Requester and gadget color. A kérdezők és a gadgetek színe.

BVLB - Bright bevel color. A pontok és a csoportokban lévő tárgyakat összekötő vonal színe.

BVLD - Dark bevel color. A kiválasztott tárgyak és pontok színe.

GRID - Grid color. A rács színe.

PICK - Picked edge color. A kijelölt vonal színe.

PPNT - Picked point color. A kijelölt pontok színe.

Preferences

SPCK - Selected, picked edge color. A kijelölt és kiválasztott élek színe.

BWLC - Line color in "B&° Wire/Shade". A vonal színe B&W Wireframe és Shade módban.

CLOC - Line color in "Color Shade". A vonal színe Color Shade módban. Csak ha a BWLN vagy COLN True.

STAR - Star color. A Globals csillagainak színe.

COLO - Color 0 value. A 0-s színregiszter színe.

NUMS - Number of screens for Make Movie. A képernyők száma, amelyeken az animációt összerakja a program.

EDLE - Antialiasing level. Az antialiasing mértéke. 255 a legkisebb, 0 a legnagyobb.

RSDP - Resolve depth. Fényútszimuláció mértéke. Minél nagyobb ez az érték, az egymásban tükröződő felületeket annál finomabban készíti el a program.

GNDN - # of divisions in ground. A preview ablakban megjelenített ground ennyi rácselemből áll.

OCTD - Number of Octree levels. Az Octree egy algoritmus, amely gyorsítja a számítást, leegyszerűsítve a tárgyakat annak eldöntésekor, hogy érhető-e fény, vagyis, hogy látható-e a felülete. A nagyobb Octree gyorsítja a renderinget, de több memóriát igényel és tovább tart az inicializálás. Gyors preview-nál tehát nem hasznos a magas érték.

GNDS - Ground size. A ground végtelen sík ugyan, de a számítás gyorsítása miatt a program korlátozott mérettel számol. Ha megnöveled a Globals méretét, ezt is növeljed meg, ellenkező esetben a ground nem látszik, majd végtelen méretűnek.

SCRL - Scroll percentage. A szerkesztő ablak ennyi százalékkal mozdul el a kurzor gomb használatával.

GSIZ - Grid size. A rács alapértelmezés szerinti mérete.

Preferences

QINT - Quick render light intensity. A lámpa fényereje quickrender módban.

EYES - 3D Stereo eye separation. 3D Stereo módban a két kamera egymástól mért távolsága.

OTRL - May Octree RAM. Az Octree számára fenntartott RAM mérete kB-ban, vagy -1, ha nincs ilyen RAM.

OTFL - Min. free RAM after Octree. A minimálisam meghagyandó RAM mérete az Octree lefoglalása után.

Nos, ennyi az általános konfiguráció. Következő a **Rendering Presets**. Itt állíthatjuk be, hogy milyen előre definiált képernyő módok és méretek jelenjenek meg a Preset List-ben. Egy bejegyzés a következőképpen néz ki:

Text - A Preset List-ben megjelenő név.

Width - A kép szélessége, maximum 8192 pixel.

Height - A kép magassága, maximum 8192 pixel.

XAspect, YAspect - A pixelek oldalaránya, általában a megjelenítő eszköztől függ.

Viewmodes - Megjelenítési mód. Lehet HAM, HIRES és LACE akár kombinálva is.

Ezeket az értékeket az alul lévő inputmezőben adhatjuk meg. Ne felejtse el a változtatásokat enterrel lezárni!

A következő csoportja, a konfigurációs lehetőségeknek, a funkcióbillentyűkhöz rendelhető parancsok. Húsz funkcióbillentyűt lehet használni, az első tízet önállóan, a második tízet Shiftelve. A Detail, Forms, Cycle, Stage és Action editorok rendelkezhetnek saját funkcióbillentyű-definíciókkal. A menüket egy háromjegyű hexadecimális számmal azonosítjuk. Az első jegy a menü sorszáma, balról-jobbra, egyel kezdődően. A második jegy a menüpont száma, a legfelső az egyes. Ha van

Preferences

almenü, azt a harmadik számjeggyel adhatjuk meg. Hogy később is tudjuk, mit állítottunk be, megjegyzéseket lehet fűzni a definíciókhoz.

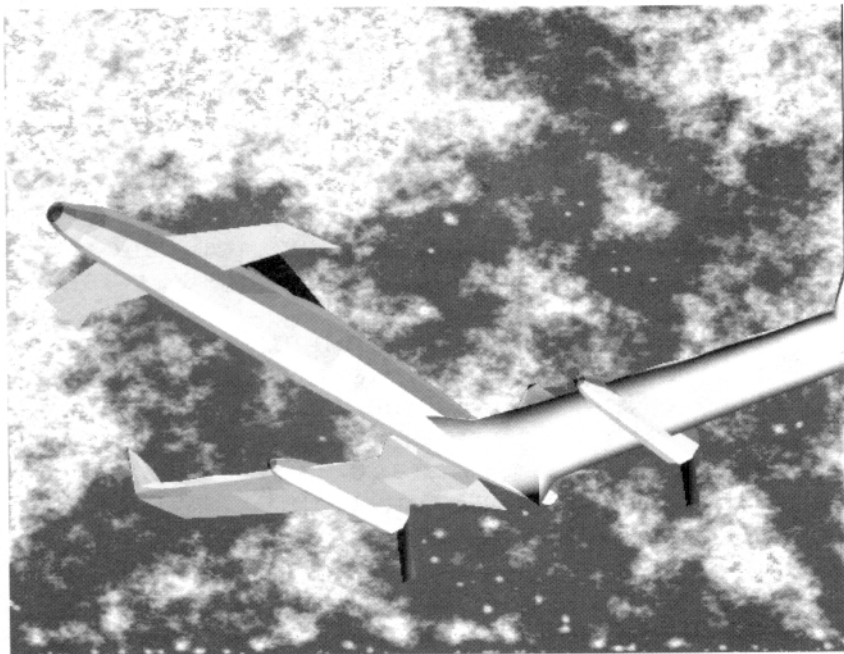
Mind a menüdefiníció, mind a megjegyzés az alsó két inputmezőben adható meg.

Elérkeztünk az utolsó szakaszhoz, a felhasználói gadgetek definiálásához. Ezek nagymértékben megkönnyítik a program használatát. A gadgetek meghatározása is három részből áll.

Text - A gadgetek felirata. Igyekezz rövid neveket kitalálni, ugyanis csak egy sornyi gadget hozható létre.

Menü # - A kapcsolóhoz rendelt menüpont meghatározása, az előző módon.

Comment - No Comment



Gyakoribb hibák és okaik

A program használatakor gyakran találkozhatunk olyan esetekkel, amikor az eredmény nem az elvárásoknak megfelelően alakul. Ezek általában nem a program hibáiból erednek, bár vannak olyanok is. A most következő részben összegyűjtöttem a hozzám érkező kérdésekből és saját tapasztalataimból a leggyakoribb hibákat és megpróbáltam azokra magyarázatot találni, elkerülésükre tanácsot adni. A felsorolás nem lehet teljes, ha olyan kérdésed van, amelyre nem sikerül megtalálni a megoldást, bátran írd egy válaszborítékkal kiegészített levelet a kiadó címére, ha tudok, segítek.

A leggyakoribb kérdések a sebesség növeléséhez kapcsolódnak. Bármilyen gyors gépen dolgozunk is, az mindig lassúnak fog bizonyulni. A gép méggyorsabbra cserélésénél olcsóbb, ha megpróbálunk okos szervezéssel gyorsítani a program futásán. A preview képek készítésénél a Preferences-en keresztül csökkentsd le az antialiasing mértékét, valamint a Resolve Depth-et. Ez jelentős gyorsulást eredményez.

Szinte senki nem gondolná, de a tárgyak mérete (nem a fájl méret, hanem a fizikai) is hatással van a sebességre. Az extrém kicsi tárgyak, amelyek mérete néhány Imagine egység nagyon lelassítják a renderinget, ezért kerüld a használatukat, inkább az egész scene-t nagyítsd fel.

A tárgyak mérete mellesleg minden irányban 1/65535-től 65535-ig terjedhet.

A fényforrások számának csökkentése csökkenti a rendering time-ot is. Gyakorlatilag két-három fényforrás elegendő a scene megvilágításához. Az

Gyakoribb hibák és okaik

egyiket helyezd jó magasra, ez lesz a nap, a másik a kamera mögül, annál kissé magasabbról világítsa meg a tárgyat. Persze ez nem általános érvényű elrendezés, az adott scene dönti el a pontos beállítást.

A tárgyak részletessége szintén befolyással van a rendering idejére. Azokat a tárgyakat, amelyeket a kamerától távolabb kívánunk elhelyezni, elég kisebb részletességgel elkészíteni, úgysem fog érvényesülni a finomabb kidolgozás.

Ha már a tárgy tervezésekor eldöntötted a beállítást, a várhatóan nem látható részek el is hagyhatók. Ez még memória és tervezési idő megtakarítással is jár.

Vannak olyan animációk, amelyek háttere mozdulatlan, de a képek kinézete miatt ezt is ray-tracelni kell, ami igen csak lassítja a dolgokat. Ebben az esetben külön készítsd el a háttérrel, majd a többi képnél használd fel Backdrop Picture-ként, vagy utólag másold egybe minden képkockával.

Ray-tracing módban hiányzik a kép egy része, miért? A program gyorsabb Képszámítás miatt trace módban csak egy tengelyenként 1024 egység sugarú, 0,0,0 középpontú gömbön belül számol. Ami ezen kívül esik az nem látszik. A fényforrások és a kamera kívül lehet ezen a területen.

A figyelembe vett méretet a Global Size felsorolójelében állíthatjuk nagyobbra. Vigyázz, a nagyobb méret hosszabb számítási idővel járhat!

Az átmásolt tervet nem lehet használni, a rendering hibaüzenettel leáll. A Staging fájl a színpadon lévő tárgyakat általában teljes elérési útjukkal együtt tárolja. Ha a terv máshová kerül, nem találja meg a program a fájlokat. Ezen úgy

Gyakoribb hibák és okaik

segíthetünk, hogy az Actionban az Actor felsoroló-jelekben átírjuk az elérési utakat.

Okozhatja a hibát a textúrák, brus-ök, effektek megváltozott elérési útja is.

Az Assigng parancs megkönnyítheti a munkánkat, a terv tartalomjegyzékét logikai egységgé lehet tenni, így a hosszú elérési út helyett, elég a logikai egység-nevet beírni.

A program floppyról történő használatakor találkozhatunk az "**Error opening staging**" file hiba-üzenettel a Stage vagy Action editorokba való belépéskor. A hiba oka legtöbbször, hogy nem az a lemez van a meghajtóban, amelyiken a terv van. Ha a program overlay módban dolgozik, azaz editor váltáskor tölti be a szükséges programrészt, a program lemezének valamelyik drive-ban kell lenni. Egy floppy esetén ne használd ezt az opciót, mert akkor a terv csak a programlemezen lehet.

Az elkészült képek minősége gyenge. Kicsi volt az aliasing mértéke vagy a fényűtszimuláció (Resolve Depth). A program mindig 24 bites képeket készít, majd azt konvertálja le az altervnel beállított színmélységűre. A konverter rutin nem a legjobbak közül való, ezért érdemesebb 24 bites képeket készíteni, majd utólag átalakítani azokat valamelyik jó képfeldolgozó programmal.

Amígán tudom ajánlani az **Image FX** vagy **ADPro** programokat, amelyeket Magyarországon a **Thalypex** forgalmaz. Mindkét program többféle dithering eljárásra képes a képek konvertálására, a legjobb eredményeket biztosítva. A programok ezen kívül hasznos képfeldolgozó funkciókkal rendelkeznek és AREXX-ből is vezérelhetők.

Gyakoribb hibák és okaik

A kérdezőben egy input mezőbe új értéket beírva és az OK-ra klikkelve a változtatás nem jelenik meg. Az input mezőkbe beírt értékek csak akkor maradnak meg, ha beírásuk után leütjük az entert. Ha ez elmarad, úgy klikkelünk az OK-ra, vagy másik input mezőre, a régi érték marad meg.

A Stage editorban valamelyik tárgyat manipulálva az nem marad úgy. Az adott kockában nincs a tárgynak a manipulációnak megfelelő felsoroló jele. Vagy adni kell egyett az Actionban, vagy használni kell a Position, Alignment, Size Bar menüpontokat.

Hiába magas a tárgy filter értéke, a tárgy mégsem látszik át. A tárgy Shininess attribútuma nem nulla, ezért a Filter nem az átlátszóságot szabályozza, hanem a fémes tükröződést.

Nem elég fémes a tárgy kinézete. A fémességet nem a Reflect, hanem a Shininess szabályozza.

A kép trace módban készült, mégisincsennek árnyékok. Túl nagy az ambient light a Globals-ban. A színösszetevőnként 25-ös érték már magas lehet.

Az animáció darabos, ugrál. Az animációs ciklusok rosszul vannak beállítva. Például egy körbeforduló tárgy az első és az utolsó kockában is ugyan abban a 0°-os pozícióban van, ezért az animáció ismétlésekor a forgás megtorpan.

Másik ok lehet, hogy a lejátszás sebessége alacsony. Ahhoz, hogy az animáció élvezhető legyen minimum 15 frame/s lejátszási sebesség kell. Nagyobb képméretnél, főleg PC-n flick anim esetén, ahol az animáció folyamatosan töltődik a hard

Gyakoribb hibák és okaik

diszről, sokszor nem lehet tartani ezt a sebességet.

Az igazán szép animációhoz 25-50 frame/s sebesség kell, amelyet kevés számítógép tud produkálni nagy felbontású, egész képernyős, hosszú animációk esetén.

Mekkora brush-t használjunk? A tárgyra kerülő brush méretét és felbontását az határozza meg, hogy az mekkorában fog látszani.

Például ha egy 640*512 pixeles képet kitölt egy olyan falfelületet, amelyet egészen letakar a brush, a felfeszítendő képnek szintén 640*512 pixel körüli méretűnek kell lenni. Kisebb brush esetén csúnyán látszani fognak a pixelek, míg nagyobb kép nem ad több információt, de jelentős memóriát köt le a rendering folyamán.

Ellentétben ha az előbbi falon csak egy akkora területet foglal el a brush, ami a végső képen 40*40 pixeles lesz, elég ha a brush is 40*40-es.

A gyorsan mozgó tárgyak a fotókon bemozdulásos életlenséget mutatnak, lehet-e ilyen képet készíteni az Imagine-nel? Közvetlenül nem, de egy trükkel igen. Készíteni kell egy kb. 10 kockás minianimációt, amelyben a bemozduló tárgy csak nagyon picit mozdul. A képkockákra egy képfeldolgozó programmal, például az ADPro-val egymásra kell másolni, úgy, hogy az első intenzitása 100% , a másodiké 90%, stb. A képeket a helyes mozgásirány kialakítása miatt fordított sorrendben kell felhasználni. Mivel a mozdulatlan tárgyak minden képen azonosak, azokon nem történik változás, ellenben a mozgást végző tárgyak az egyre kisebb intenzitású képeken bemozdulni látszanak. A munka eredménye egyetlen kép, amelyen bemozdulásos életlenség lesz.

Gyakoribb hibák és okaik

A fenti eljárás az Imagine világhírű "nagy öregétől", **Steve Worley**-től származik. Ő készítette vele a **Bowling**, címet viselő képet.

Hogyan lehet láthatóvá tenni a fény útját? Magát a fénysugarat nem lehet megjeleníteni, de egy kis csalással hasonló eredményt érhetünk el. A Detalil editorban készíteni kell egy olyan alakú tárgyat, mint a fénycsóva (henger, vagy kúp), tengelyét a csúcsba tenni és a tengelyek hosszát úgy beállítani, hogy az Y tengely végigérje a kúpot, vagy hengert annak középvonalában, az X tengely pedig az alap sugarával azonos méretű legyen. A tárgyat el kell látni fényforrás attribútummal és ezt kell a fény-sugár helyére tenni. A tárgy attribútumai közül a színt kell beállítani a fénynek megfelelőre és a Fog lenght-et a fénycsóva átmérőjének kétszeresére. A halvány köd kinézetű tárgy jól modellezi a fény útját.

Hogyan készítsünk élethűbb csillagos eget? Az Imagine Star field density funkciója nem igazán élethű csillagos eget állít elő. A csillagok mindig egy pixel méretűek, és azonos színűek. Célszerű inkább egy csillagos égbolt fotót gömbre feszíteni és a színpadot gömbön belül felépíteni.

Hogyan lehet élethűbb tárgyakat készíteni? Én személy szerint azt a módszert követem, hogy vásárolok egy makettet, minél nagyobb méretarányban és tapintókörző valamint tolómérő segítségével leveszem róla a méreteket, amelyeket azután Imagine egységbe átalakítok és úgy építem fel a tárgyat.

A modellt alkotó darabokat külön-külön elkészítve, azokat csoportba kötve áll össze a végső tárgy.

Záróra

Minden jónak vége szakad egyszer, így ennek a könyvnek is. Remélem azonban, hogy sikerült bevezetni mindenkit a ray-tracing világába, érthetőbbé tenni az Imagine-t. Mi mást kívánhatnék legvégül?

Kívánom, hogy rövid időn belül légy mestere a tárgy és képkészítésnek, munkád nyomán csodálatos képek és animációk szülessenek.

Essen szó azokról is, akik nélkül e könyv nem jöhetett volna létre. Köszönöm családtagjaimnak, barátaimnak a munkám során nyújtott segítségüket. Mindenkit felsorolni hosszú lenne, de külön is köszönettel tartozom Böbének, Gáborcának és Alexnek.



Tiszaföldvár, 1994. 02. 25.

