



Felületi mintázatok a Blenderben (5. rész)

Ebben a részben megtanuljuk, hogyan lehet valódi tükröző felületet létrehozni a Blender segítségével.

Kezdeném talán a legegyszerűbb résszel, vagyis a képek és animációk mintázatként való felhasználásával. Állítsuk be a tárgy anyagát, kapcsoljunk át az F6 billentyűvel a mintázatok létrehozására alkalmas nézetbe. Itt a korábbiakban még nem ismertetett *Image* mintázatot kell választanunk, és az alábbi leírás alapján az igényeknek megfelelően kell majd beállítanunk a tulajdonságokat. A mintázat típusa alatt látható gombsorral kezdetjük az ismerkedést. Az *InterPol* kapcsolót arra használhatjuk, hogy amikor a tárgyra ráfeszítjük a mintázatot, a tárgy és a felhasznált kép mérete nem minden esetben azonos, tehát a képet a tárgy teljes befedéséhez a legtöbbször nagyítani kell. Ilyen esetekben a kép pontjait nem lenne célszerű csupán többszörözni – a szebb látvány eléréséhez két képpont közé színátmenetet képezhetünk. Ha például egy képet a kétszeresére nagyítunk, akkor egy piros és egy fehér képpont közé nem egyszerűen egy újabb piros pontot helyezünk el, hanem a két pont átlagát, vagyis egy rózsaszínű képpontot. Természetesen, ha több képpontot kell beilleszteni, akkor a következő képlet alapján számíthatjuk ki a megfelelő szint:

forrás_képpont+pillanatnyi_beillesztendő((cél_képpont-forrás_képpont) / beillesztendő_pontok_száma)*

Ezt nevezzük lineáris interpolációnak, ennek megvalósítására alkalmas az *InterPol* kapcsoló.

A következő kapcsoló a *UseAlpha*, ami arra jó, hogy a Blender az átlátszóságadatokat is tartalmazó képeken figyelembe vegye ezt a tulajdonságot. Ilyen képeket könnyedén előállíthatunk például a Gimpel. A képet TGA vagy PNG formátumban tárolva ez az adat megmarad és használható. Amennyiben alkalmazni kívánt képünk nem tartalmazza ezt az adatot, a Blender a *CalAlpha* kapcsoló alkalmazásával képes ezeket az értékeket kiszámítani. Az átlátszóság kiszámításához a Blender a képpontok intenzitását használja fel, vagyis minél világosabb egy képpont, annál kevésbé lesz átlátszó a mintázat az előállításakor. A *NegAlpha* kapcsolóval a kiszámított vagy tárolt átlátszóságértékek inverzét használhatjuk, vagyis ebben az esetben a sötétebb képpontok lesznek jobban átlátszók, míg a világosabbak kevésbé. Természetesen a tárolt átlátszóság-adatok esetében az érték nem függ a képpont világosságától, ott teljesen szabadon határozhatjuk meg a képpontok fényáteresztő képességét.

A *MipMap* kapcsolóval elérhetjük, hogy a Blender a képpontokat ne csupán két dimenzióban interpolálja, hanem a kamerától való távolság figyelembevételével több különböző felbontású képet készítsen a megadott mintázatból, és a kép kiszámítása során használja őket. Ennek az lehet az előnye, hogyha egy tárgy messzebb áll a kamerától, akkor nem kell akkora képpel számolni a megjelenítés során, vagyis gyorsabban kiszámítható a végeredmény. Viszont ahogy a tárgy közelebb kerül, a Blender a nagyobb felbontású mintát használja, tehát a tárgy mintázatának részletei is megjelennek. Ennek jó ellenpéldája

a réges-régi Wolfenstein 3D program, ahol a falhoz közeledve a falminta nem részletesebbé, inkább élvezetlenebbé vált. A *Rot90* kapcsolóval a mintázatként használt képet 90 fokkal forgathatjuk el, míg a *Movie* kapcsoló segítségével közölhetjük a Blenderrel, hogy nem egyszerű képről van szó, hanem egy animációról, ami képkockáról képkockára változik majd a tárgyon. Amikor animációt szeretnénk felhasználni, akkor a nézet közepén található *Frames* értékkel állíthatjuk be, hogy a kiszámított animációban mennyi képkockán keresztül kell ezt a mintázatot figyelembe venni, míg az alatta található *Offset* értékkel azt határozzuk meg, hogy azt a Blender a videórészlet hányadik képkockájától kezdve jelenítse meg a tárgy mintázataként.

A *Anti* kapcsolóval kapcsolhatjuk be az élek finomítását, amivel – hosszabb számítási időért cserébe – szebb eredményt kaphatunk.

A fentiekben tárgyalt kapcsolók alatt találjuk az egyik legfontosabb gombot *Load Image* felirattal, amivel megadhatjuk a Blendernek, hogy melyik képet vagy animációt szeretnénk a mintázat létrehozásához felhasználni. A gomb használatával egy állománylista jelenik meg, ahol kiválaszthatjuk a megfelelő képet, és a *Load* gombbal betölthetjük. Amennyiben olyan animációt adunk meg, amit a Blender nem képes betölteni, hibaüzenet figyelmeztet minket. Tapasztalataim szerint a Blenderrel készült animációk betölthetők, emellett a tömörítetlen képkockákat tartalmazó avi formátumú és a Motion JPEG tömörítéssel készült avi formátumú videórészletek használhatók. A kép vagy animáció betöltése után a fájl elérési útja megjelenik a gomb melletti mezőben, és az állománynév mellett található *Reload* gomb használatával a Blender újra beolvassa az állományt. Ez a lehetőség akkor hasznos, amikor a pontos mintázat még nem készült el, vagyis folyamatos módosítás alatt áll, és szeretnénk megtekinteni a módosítások eredményét. A képek betöltésére szolgáló gomb alatt találhatjuk a *Filter* értéket, amivel azt határozhatjuk meg, hogy az interpolált nagyítás és az előbbieken tárgyalt *MipMap* szolgáltatás használatakor a Blender mekkora környezetet vegyen figyelembe a köztes értékek kiszámításakor.

A következő sorban lévő kapcsolókkal azt határozhatjuk meg, hogyha a kép vagy animáció kisebb, mint a tárgy, akkor a Blender hogyan számolja ki a tárgy fényáteresztő képességét. Az *Extend* kapcsolóval a kép szélén található színértékek a tárgyra is kiterjednek majd, a *Clip* hatására a mintázat által le nem fedett területek átlátszósága nulla lesz, a *ClipCube* használatával a kép köré rajzolható kockán kívüli területek átlátszósága lesz nulla értékű, míg a *Repeat* alkalmazásával a kép vagy animáció átlátszóságértékei is megismétlődnek a tárgyon. E kapcsolók alatti *XRepeat* és *YRepeat* értékekkel állíthatjuk be, hogy a mintázat a tárgyon a vízszintes és a függőleges tengely mentén hányszor ismétlődjön. A képek mintázatként való felhasználásakor még azt is fontos meghatározni, hogy a Blender mekkora területet vegyen figyelembe a tárgy mintázataként. Erre szolgál a *MinX*, *MinY*, *MaxX* és *MaxY* érték,

ezekkel pontosan beállíthatjuk a mintázat határait, és ezzel a megoldással elérhetjük azt is, hogy egy képen több tárgy mintázatát is megrajzolhassuk.

Nos, ennyi talán legendő ahhoz, hogy tetszőleges képet helyezhessünk el egy tárgyon, meglehetősen pontosan és változatosan. Most ismerkedjünk meg azokkal a lehetőségekkel, amelyek segítenek a tükröző felületek előállításában. Azoknak, akik figyelemmel kísérték a sorozat korábbi részeit, talán feltűnt, hogy a mintázatok tárgyalásából kimaradt az **EnvMap** típusú felületi mintázat. Itt az alkalom, hogy ezt a hiányt pótoljam, és megismertessem olvasóimat ezzel a lehetőséggel



1. kép Egyszeres tükröződés



2. kép Többszörös tükröződés

és az eredményül kapott kép látványának nagyszerűségével. Az érthetőség kedvéért itt is az egyszerűbb látvánnyal kezdem. Amikor a Blendert elindítjuk, a jelenetben alapértelmezetten találunk egy tárgyat, használjuk fel ezt a tanulás során. Hamarosan ez a síkfelület lesz a tükröző felületünk. Hozzunk létre egy lámpát és egy gömböt, hogy legyen valami, ami tükröződik a síkon. A síklapot az **ALT-M** hatására megjelenő gombsor segítségével helyezzük át a második rétegre, majd mindkét réteget tegyük láthatóvá. Ezután jelöljük ki a síkot, és hozzunk létre egy új anyagot és egy új mintázatot is. A mintázat típusaként az **EnvMap** kapcsolót kell használnunk. Ez az a pont, amikor meg kell értenünk, hogy a Blender mi módon képezi le a környezetét visszatükröző tárgy mintázatát. Amikor egy ilyen mintázatot létrehozunk, a Blender egy virtuális kamerát helyez el a megadott ponton, és mindhárom tengellyel párhuzamosan két-két képet számít ki a jelenetről. Hármát pozitív irány felé néző kamerával, további hármát negatív irányba tekintő kamerával. A későbbiekben ezeket a képeket fogja használni mintázatként, mégpedig olyan módon, mintha egy kockába zárt volna a tárgyat, aminek oldalain ezek a számított képek láthatóak. Az előbbieken alapján már látható, hogy a képek elkészítéséhez meg kell adni egy térbeli pontot, itt fogja a program elhelyezni a virtuális kamerát. Amikor a **EnvMap** kapcsolóval meghatároztuk a mintázat típusát, a nézetben megjelent egy **OB:** mező, amiben meg kell adni azt a tárgyat, ami ezt a pontot meghatározza. Azért, hogy e térbeli hely ne befolyásolja majd az összképet, a **Főmenü/Add/Empty** menüpontok kiválasztásával célszerű egy üres objektumot létrehozni. Ebből a pontból számítja ki tehát a program a képeket, ezért az előbbi **OB:** mezőbe írjuk be a látszólagos tárgy nevét (a tárgy neve „Empty”). A kamera elhelyezése után nyomjuk meg a **SHIFT-N** billentyűt, aminek a hatására megjelennek a kiválasztott tárgy tulajdonságai (ebben az esetben a kamera legyen a kiválasztott tárgy). Jegyezzük meg (írjuk fel) ezeket az értékeket, mert a helyes tükrökép előállításához a referenciaobjektumot pontosan a kamera alatt kell megjelölni, csak a **Z** koordinátát

kell ellentétes előjellel alkalmazni. Ez a megoldás akkor működőképes, ha a síklapot **Z** irányban nem mozgattuk el. Amennyiben elmozdítottuk, akkor a térbeli pont és a kamera közötti függőleges távolság felénél kell lennie, tehát a pont vagy a kamera magasságát ennek figyelembe vételével újra kell számolni. Feltételezve, hogy a síklapot nem mozdítottuk el, jelöljük ki a tükröződés számításához szükséges pontot, majd a **SHIFT-N** billentyűkombinációval állítsuk be a térbeli helyét. Az **X** és **Y** koordinátáknak meg kell egyezniük a kamera **X** és **Y** koordinátaival, a **Z** koordináta pedig legyen ugyanakkora, mint a kamera és a síklap távolsága, de azzal ellentétes előjellű (az

értéke alapesetben megegyezik a kamera **Z** koordinátájával, az előjele azonban negatív). Ezek után már csak azt kell a program tudtára adni, hogy ezen a tárgyon valóban használni is szeretnénk a tükröződéshez kiszámított mintázatot, tehát jelöljük ki a síklapot, kapcsoljunk át az anyagszerkesztő nézetbe (**F5** billentyű), és a nézet közepén kapcsoljuk be a **Ref1** kapcsolót. Nézzük meg, mire jutottunk eddig: az **F12**-vel számíttassuk ki a képet. Láthatjuk, hogy a síklapon nem jelent meg a tükröződő gömb. Ez azért történhetett így, mert amikor a program a fentebb említett hat képet kiszámította, a tükrö-

zódás kiszámításához használt nézőpontból a síklap eltakarta a gombunkat. Természetesen erre is létezik megoldás. Kapcsoljunk vissza a mintázatokhoz, ahol a **Don't render layer:** felirat alatti gombsoron beállíthatjuk, hogy a program a számítások során milyen rétegeket hagyjon figyelmen kívül. Itt kapcsoljuk be a második réteget, hiszen a jelenet létrehozásakor a síklapot erre a rétegre helyeztük át. Ha most a végeredményt egyszerűen újból kiszámítanánk, az nem változna semmit, mert a szükséges képeket a program már az előbbi számolás során előállította. A korábban kiszámított adatokat töröljük a **Free Data** gomb használatával, és számoljuk ki újra a végeredményt. Ebben az esetben, ha mindent jól csináltunk, a síklapon meg kell jelennie a gömb tükröképének, ami azt jelenti, hogy továbbléphetünk a következő látvány megalkotásához. Célunk most az lesz, hogy a síklapon két gömb tükröképét lehessen látni, s ezek közül az egyiket a másik gömb is visszatükröződjön. Hozzunk létre egy újabb gömböt a kamera látóterén belül, és a térbeli kurzor áthelyezése nélkül adjunk meg egy üres objektumot is. Az új gömböt helyezzük át a harmadik rétegre, és ezt is jelenítsük meg. A gömbnek adjunk anyagot, és mintázatként állítsuk be az **EnvMap** típust. Az anyagtulajdonságok között ne felejtjük el bekapcsolni a **Ref1** kapcsolót. Tekintettel arra, hogy „Empty” nevű objektum már szerepel a jelenetben, ezt a nevet már nem adhatjuk meg kiindulási pontként a tükröződés számításához. Jelöljük ki az utoljára elhelyezett üres tárgyat (ez most a gömb belsejében található), és az **F9** billentyűvel kapcsoljunk át a tárgyszerkesztő nézetbe. Itt a fejlécsről leolvashatjuk a tárgy nevét (az eddigi lépéseket követve ez **Empty.001** lesz), amit az újabb gömb mintázatának kiindulási tárgyaként be kell írunk az **OB:** mezőbe. Utolsó lépésként már csak azt kell megadnunk, hogy a Blender milyen rétegeket hagyjon figyelmen kívül a számítások során. Kapcsoljuk be a harmadik réteget, amin az új gömb található, hiszen az nem átlátszó – a tükröződés számításához használt képeken a gömb belseje jelenne meg. Ezek után a programmal számíttassuk ki a végeredményt.

SHIFT-N	Helyi koordináták és elfordulásértékek
ALT-M	Tárgy áthelyezése másik rétegre
F5	Anyagtulajdonságok
F6	Mintázatok
F9	Tárgyszerkesztő nézet
F12	Kép kiszámítása

Láthatjuk, hogy ugyan a síklapon mindkét gömb tükörképe megjelent, a második gömbön szintén látható az első tükörképe, de mintha valami még nem lenne tökéletes. A képet jobban szemügyre véve feltűnik, hogy a második gömb tükörképén nem tükröződik az első gömb. Ennek az az oka, hogy a Blender a számítások során nem használ sugárkövetést, vagyis első lépésben nem számítja ki az egynél többszöri tükröződések. Ezt láthatjuk az 1. képen. E nehézség megoldására „trükközödést” használunk. Miután ezt a kissé hibás képet kiszámítottuk, még egy lépést meg kell tennünk: töröljük a síklap kiszámított adatait – F6 → *Free Data* –, majd számítsuk őket újra. Így a második gömbről készülő képen már látható az első gömb képe, vagyis a kívánt végeredményt kapjuk meg, ezt láthatjuk a 2. képen. További tükörtrükköket az ismertetett két módszer használatával készíthetünk. Ne feledjük el megadni a megfelelő kiindulási tárgyat, sem a megfelelő tárgyakon kétszer kiszámítani a környezetről készülő képeket, továbbá az anyagtulajdonságoknál bekapcsolni a Refl kapcsolót. A gyakorlás során biztosan hamar feltűnik nekünk, hogy a tükörképek, bizony, meglehetősen alacsony felbontásban

készülnek, azaz nem túl szépek. Természetesen a Blender erre is kínál megoldást, tekintsük csak meg a mintázatszerkesztő CubeRES értékét. Ennek alapbeállítása 100, ami nem túl nagy. Előfordulhat, hogy a tárgy–kamera-távolság olyan nagy, hogy a kis felbontású számítások is kielégítő eredményt adnak, de ha mégsem, akkor ezt az értéket szívünk és igényeink szerint legfeljebb 1000-re növelhetjük. Ezzel azt határozzuk meg, hogy a környezet leképezésekor használt kocka egy-egy oldalát alkotó kép milyen méretű legyen. Ez az érték képpontokban értendő. Amennyiben egy teljes animáció során szeretnénk a környezet leképezését alkalmazni (vagyis tükröződő felületeket számoltatni a programmal), a mintázat nézetében található Static kapcsolót kapcsoljuk át a mellette látható Anim elnevezésűre. Így a Blender minden képkocka kiszámítása előtt létrehozza majd a környezet mintázatát.

Végezetül a táblázatban láthatók a használt billentyűkombinációk, az újak és a már ismertek egy része egyaránt.

A CD-mellékleten (48. CD Magazin/Blender) található példákat tanulmányozva a Blender és a tükröződő felületek kapcsolata remélhetőleg könnyen érthető. Természetesen legjobb tudásom szerint szívesen válaszolok levélben feltett kérdésekre is. Legfontosabb tanácsom: továbbra se mulasszuk el folyamatos gyakorlást.



Fábán Zoltán (dzooli@freemail.hu, dzooli@yahoo.com) 27 éves, jelenleg programozóként dolgozik. Szabadidejében szívesen kirándul, túrázik. Emellett szeret rajzolni, érdeklődik a 3D grafika és a Linuxszal kapcsolatban minden olyan program és programnyelv, amit még nem ismer vagy nem próbált ki.

