



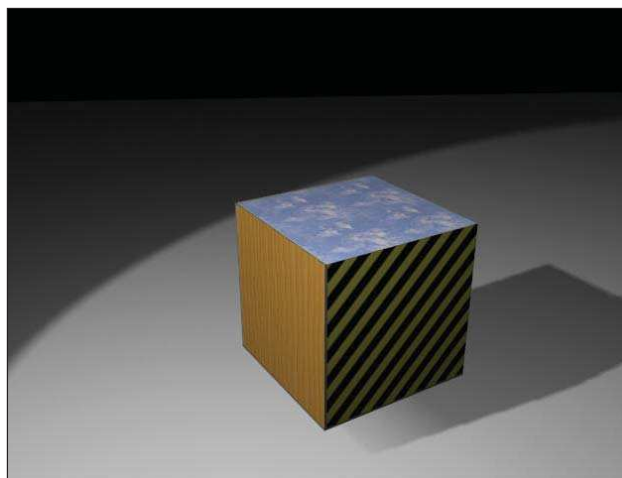
Felületi mintázatok a Blenderben (3. rész)

Sorozatunk e részében egy olyan hatékony eljárást szeretnék bemutatni, amivel egy adott tárgyon pontosan a kívánt helyzetben helyezhetők el a mintázatok.

Ennek az eljárásnak a neve angol szakkifejezéssel UV-mapping, amit UV-térképezésre fordíthatnánk le. A módszer alkalmazása során a Blendernek pontosan meg tudjuk adni, hogy a tárgyat alkotó elemek csúcspontjaiban (háromszögek, négyszögek) a mintázatként használt kép melyik pontja jelenjen meg. Ezt az eljárást fogjuk majd felhasználni a Blender játékmotorjának a megismerése során. Játékfejlesztéskor pedig különösen elterjedt megoldás, hogy a háromdimenziós objektumokat kevés alkotóelemből építik fel (ez úgynevezett low-poly modelling, azaz alacsony poligonszámú modellezés), ám annál nagyobb gonddal dolgozzák ki a mintázatokat, ezzel érve el a gyors megjeleníthetőség mellett is szép hatást. Az egyszerű megoldással kezdeném az ismertetést, első megközelítésben egy kockának minden oldalát különféle mintával borítjuk be. Még mielőtt a Blendert munkára fognánk, készítünk el a mintázatot alkotó képeket. Mivel a tárgyat alkotó elemek helye pontosan ismert, ezt „kézzel” is megrajzolhatjuk. Én az XFig programmal készítettem egy négyzetet, majd még öt példányban lemásoltam, ahogyan az általános iskolai tanulmányaim során a papírkocka hajtogatásakor tettem. Ezt a kiindulási képet PNG formátumba exportáltam, majd a Gimpel kifestettem. Kiválasztottam a megfelelő négyzet belsejét, és mintával töltöttem ki mind a hat négyzetet. A gyakorláshoz el sem kell készíteni őket, ugyanis ezek a képek a CD-mellékleten is megtalálhatók. A képek elkészülte után indítsuk el a Blendert.

Elsőként hozzunk létre egy új kockát a *Főmenü-Add-Mesh-Cube* menüpontok kiválasztásával. Miután a kocka megjelent, az A billentyűvel szüntessük meg az összes oldalának kijelölt állapotát, és a TAB-bal kapcsoljunk át tárgyszerkesztő módba. Az UV-térképek elkészítéséhez át kell váltanunk *Vertex-paint* módba. Az alsó panelen látható egy rádióaktivitást jelző kis ikon. Nem kell megijednünk tőle, viszont a váltáshoz a tőle balra lévő gombot kell használnunk, ami leginkább egy kis ecsethez hasonlítható. A következő lépésben – a már ismert módon – a szerkesztőnézetet osszuk kétfelé, és a jobb oldali nézetben nyomjuk meg a SHIFT-F10 billentyűket. Ezzel a nézetet átváltottuk az UV-szerkesztő módba, és a *Load* gomb alkalmazásával betölthetjük az előre elkészített képet. Megjegyzendő, hogy a Blender csak a PNG, a JPG és a TGA formátumú képeket jeleníti meg, és ezeket a formátumokat képes felhasználni a szerkesztőben is. Tehát a kép betöltése után elől-, oldal- vagy felülnézetben ki kell választanunk azokat a háromszögeket, amelyeken majd a mintázatot látni szeretnénk. Ezt a legkönnyebben úgy tehetjük meg, hogy átkapcsolunk pontszerkesztő módba, és a nézet csekély elforgatása után kiválasztjuk a megfelelő csúcspontokat. Ezután váltsunk vissza az előző nézőpontba, és nyomjuk meg ismételtén a TAB gombot, hogy a Blenderrel elkészíthetessük a kijelölt háromszögek kiterített képét. Kijelölt kockánkon használjuk a U billentyűt, és a megjelenő menüből válasszuk ki a *Cube*, a *Standard1/1* vagy a *Window* → *From Window* menüpontot. Bármelyik mellett döntünk is, a továbbiakban az elkészített hálólal a jobb

oldali nézetben kell majd dolgoznunk. Ebben az egyszerű esetben a háló csupán egy négyzetre egyszerűsödik, hiszen a kockának az ebből a látószögéből alkotott képe egy négyzet lesz. Bonyolultabb formák esetén az elkészített háló úgy képzelhető el, mintha a kamera lencséje sík lenne, és erre terítené ki a Blender a kijelölt háromszögeket. Tehát a feladat annyi, hogy a Blender által kiterített háló pontjait a mintázatként felhasználni kívánt kép megfelelő pontjaira kell igazítani. Ebben a példában a képen a négyzet csúcsait azokra a pontokra igazítjuk, ahol a rajzolás során az oldalak határait megrajzoltuk. Ezek után a tárgyat ábrázoló nézetben kapcsoljuk ki a háromszögszerkesztő módot az F billentyű használatával. Mivel a kocka minden oldalára különféle mintázatot terveztünk, a fenti lépéseket a fennmaradó öt oldalra is el kell végezni, hogy az eredmény megfeleljen a tervben foglaltaknak.



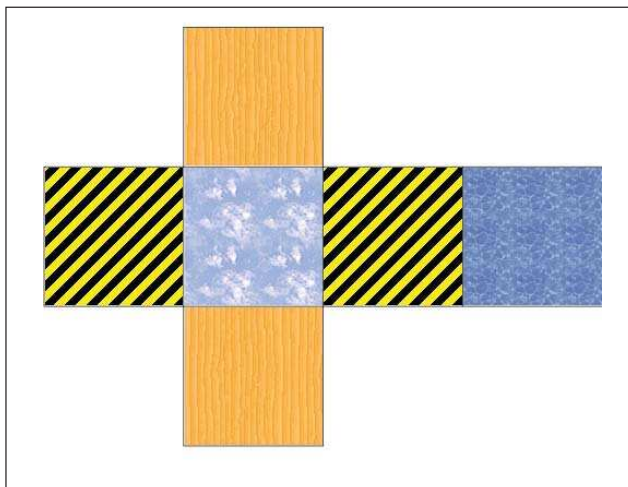
1. kép Munkánk eredménye

Ennyi munka után végül tekintsük meg az eredményt! Adjunk fényforrásokat a jelenethez, majd számoltassuk ki a képet. Láthatjuk, hogy a kockánk teljesen szürke. Adtunk-e neki valamikor anyagot? Eddig még nem, tehát itt az ideje ennek is. Kapcsoljunk az anyagszerkesztő nézetbe, a kockához pedig rendeljük hozzá az alapértelmezett anyagot. Ha most a képet újra kiszámítanánk, semmi változást nem láthatnánk, hiszen amíg semmilyen anyagot nem rendeltünk a tárgyhoz, addig az alapértelmezett tulajdonságokkal bír. Ahhoz, hogy a Blender figyelembe vegye a beállított koordinátákat, kapcsoljuk be az anyag színének meghatározása alatt a *TexFace* kapcsolót. Így a beállított mintázat már a számítások során is megjelenik a tárgyon, ahogyan ez *1. képünkön* is látható. Hogyan tudjuk a mintázatokat könnyedén létrehozni? Kapcsoljuk az egyik nézetet az UV-térképező módba, míg a másikat a 3D-megjelenítés egy tengelymenti nézetére (felül-, oldal- vagy előlnézet). A F billentyűvel váltsunk UV módba és jelöljük ki azokat a háromszögeket, amelyek közel

Az uvexport.py

```
import Blender210

print "-----"
print "  UV coordinate exporter  "
print "    for Blender 2.23    "
print "  created by Zooli      "
print "e-mail: dzooli@freemail.hu"
print "version: v0.9          "
print "-----"
scene=Blender210.getCurrentScene()
for o in scene.objects:
    ob210=Blender210.getObject(o)
    if Blender210.isMesh(obj):
        print "Object: "+obj
        m210=Blender210.getMesh(o)
        print "  Filename: "+o+"_tex.uv"
        f1=open(o+"_tex.uv", "w")
        for c in m210.texcoords:
            tc = []
            tc.append(c)
            f1.write("%s %s\n" % (tc[0]))
        f1.close()
        print "  export ok."
print "All coordinates exported."
```



2. kép A kiterített kocka

merőlegesek a nézési irányra. Az F billentyűt használjuk ismételtlen, majd az U billentyű hatására megjelenő menüből válasszuk ki a „From window” menüpontot. Ezek után a térképező nézetben megjelennek azok a háromszögek, amelyeket kiválasztottunk. Mozgassuk ezeket olyan pozícióba, hogy a későbbiekben a többi háromszögünknek is maradjon hely. Ezt a folyamatot az adott nézőpontból folytassuk addig, amíg minden közel merőleges háromszögnek meghatároztuk a mintázatpozícióit. Majd váltunk nézetet és ismételjük meg a többi háromszöggel is ezt a folyamatot. Előfordulhat, hogy egy képre nem fér rá minden háromszögünk, ilyen esetekben újabb képet is megadhatunk az UV-térképező nézetben. Ezután jelöljük ki azokat a háromszögeket, amelyeknek már meghatároztuk a koordinátáit, és nagyítsuk ki az UV-

térképező nézetet az alsó fejlécén található négy részre osztott gomb használatával. Ez balról a második gomb. A „szürke +” billentyűvel nagyítsuk a térképet olyan méretűre, hogy a képernyő legnagyobb részét betöltve még minden rész látható legyen. Készítsünk képernyőmentést, majd egy rajzoló programmal vágjuk ki a szükséges részt és rajzoljuk meg a mintázatokat. Amikor a mintázatokat rajzoljuk, a kiszínezett részekkel fedjük le teljesen a Blender által készített szaggatott vonalakat, és az sem baj, ha ezeken egy kissé túlfestünk. Amikor a mintázatot újra betöltöttük a Blenderbe a csúcspontok mozgatásával pontosítsuk a háromszögek helyzetét az UV-térképező nézetben. A pontosság megállapításához nem kell mindig újra kiszámítanunk a képet, mert a ALT-Z billentyűkombinációval a szerkesztő nézetben is megjeleníthetjük a mintázatot. Ezzel a módszerrel készült mintázatok a CD-mellékleten is találhatóak.

Itt jegyezném meg, hogy az előbb ismertetett módszerrel egy – a játékfejlesztés során használt – segédeszköz is megtakaríthatunk, ugyanis ilyen koordinátákat az elkészülő modellekhez nem kell külön segédprogrammal létrehozni, ezt a modellező programon belül is megtehetjük. Játékfejlesztés során előfordulhat, hogy ezekre a koordinátákra később is szükségünk lesz, mentésüket azonban a Blender nem támogatja. Szerencsére lehetőségünk nyílik Python programokkal kiegészíteni a Blender lehetőségeit, ahogyan azt a mellékelt programlistán is láthatjuk. Ezt a programot célszerű a Blender 2.23 változatával használni, ugyanis a későbbi változatokban nem működik, mert azokban a Python-bővítményekben használható eljárások folyamatosan változnak, így a program már az első sort sem tudja végrehajtani a *Blender210* függvénykönyvtár hiánya miatt (lásd *listánkat*).

Ez az egyszerű kis kiegészítés lehetővé teszi, hogy a Blenderben létrehozott és beállított koordinátákat a továbbiakban felhasználjuk például a saját játékunkban a mintázatok elhelyezéséhez. A könnyebb használhatóság érdekében a CD-mellékleten megtalálható a program egy kissé átalakított változata, és egy jelenet, amiben csak ez a program szerepel, tehát akár azt átalakítva vagy abból a segédprogramot betöltve bárki tetszése szerint tovább alakíthatja és használhatja. Ezután még szükség lesz magára a modellre is. A Blenderben a SHIFT-F2 billentyűvel DXF formátumban menthetjük a modelleket, aminél egyszerűbbet keresni sem kell. Leendő játékunkban akár közvetlenül, akár további átalakítások után könnyedén felhasználhatjuk az így létrehozott tárgyakat.

További fontos tudnivaló, hogy eddigi tapasztalataim szerint a Blender játékmotorja csak az így elkészített UV-térképekkel ellátott tárgyakon jeleníti meg a mintázatokat, tehát tárgyalásakor ennek az eljárásnak különösen fontos szerepe lesz. Az alábbiakban rövid ízelítőt adok a Blender képességeiből a részecske-rendszerek megjelenítésének területén. A részecske-rendszerekről elegendő annyit tudnunk, hogy a szimuláció során minden részecske mozgását a program egyenként számítja ki, és a rendszer kialakításában résztvevő részecskék egymással is kölcsönhatásban állnak. Ezt a témát sajnos nem lehet teljességében bemutatni, hiszen nagyon sok múlik a személyes alkotókészségen, így itt egyetlen példán keresztül csak az alapokat tudom bemutatni.

Első lépésként hozzunk létre egy kúpot a *Főmenü/Add/Mesh/Cone* menüpontokat kiválasztva. A Blender alaphelyzetben a részecskék kiindulási irányát a tárgy háromszögeit alkotó pontok normálvektorával megegyező irányként állítja be. Ebben a példában egy szökőkúthoz hasonló hatás megalkotását szeretném bemutatni, tehát a vízcseppeknek kezdetben felfelé kell halad-

| | |
|-----------|--------------------------|
| SHIFT-F10 | UV-szerkesztő mód |
| SHIFT-F2 | Mentés DXF formátumban |
| F | Háromszög-kiválasztó mód |
| F10 | Számítás beállításai |

niuk, ezt követően lefelé. Ha a részecskéket a kúpot változtatlanul hagyva állítanánk be, akkor azok lefelé indulnának, másrészt pedig a kúp csúcsából is elszabadulna néha egy-egy részecske. Ezt elkerülendő töröljük ki a kúp csúcspontját alkotó pontot. Jelöljük ki a csúcspontot, majd az X billentyű hatására előkerülő menüben válasszuk ki a *Vertices* pontot. Így már nem adunk lehetőséget kósa részecskék felbukkanására. A másik megoldandó feladat a kiindulási irány megváltoztatása. Jelöljük ki a megmaradt pontokat, és oldal- vagy előnézetben forgassuk el őket 180 fokkal. Ehhez használjuk az R billentyűt, és a pontos forgatás megvalósításához a CTRL gomb nyomva tartásával forgassuk el a pontokat. Így a részecskék már a megfelelő irányba indulnak. Ezután hozzunk létre egy világoskék színű anyagot, kapcsoljuk be a *Halo* tulajdonságot, és az *Alpha* értéket vegyük alacsonyabbra. A szebb látvány érdekében kapcsoljuk be még a *Rings* kapcsolót is. Ezek a beállítások megfelelnek a céloknak, de kedves olvasóim természetesen szabadon kísérletezhetnek más értékekkel is.

Ami igazán fontos, az magának a részecskerendszernek a beállítása. Az F7 billentyűvel aktiválhatjuk az animációs beállításokat. A nézet közepén egy *New Effect* feliratú gombot találunk, amit most használnunk kell. A tőle jobbra lévő listából válasszuk ki a *Particles* beállítást. Ennek hatására rengeteg beállítási lehetőség tűnik fel e két gomb alatt. Kezdjük az elején: a *Tot* értékkel a rendszerben szereplő részecskék legnagyobb számát adhatjuk meg. A *Sta*, az *End* értékek határozzák meg a részecskék mozgásának kezdő és befejező idejét, az animáció képkockáit használva mértékegységül. Ezt a két értéket állítsuk 1-re és 100-ra. A mellettük található *Life* értékkel szabályozhatjuk, hogy a részecskék első nemzedéke mennyi ideig szerepeljen a rendszerben, míg a *Keys* jelzi a Blender számára, hogy hány kulcskockát számítson a teljes életciklusuk során.

A következő sorban a *CurMul* azt mutatja, hogy hányadik nemzedék adataira vonatkoznak a mellette lévő értékek. A nemzedékekről annyit kell tudnunk, hogy amikor egy részecske az életciklusa végére ér, képes újabb részecskéket létrehozni, amiket a Blender szóhasználatában gyermekeknek nevezünk. A valóságban ilyen „gyermekeket” például tűzijáték szemlélése során láthatunk. E beállítás mellett található a gyermekek által használt anyag sorszáma, a *Mult* értékkel pedig beállíthatjuk megszülető „gyermekrészecskék” arányát. Amennyiben ez az érték 0, úgy a részecske az életciklusát befejezve eltűnik, míg a legnagyobb 1-es érték azt jelenti, hogy mindig a *Child* érték (a gombsor utolsó eleme) által meghatározott számú „gyermek” születik. Az előbbi két beállítható érték között találjuk az új nemzedék életének a hosszúságát. Természetesen az új részecskéknek is lehetnek majd utódaik, ezek tulajdonságait úgy határozhatjuk meg, ha a sor elején található *CurMul* értéket növeljük, és a fentiek alapján ismételt beállítjuk a tulajdonságokat. Alább még érdekes a *RandLife* és a *Seed*, amelyekkel az életciklus hosszúságának adhatunk véletlenszerű változást. A következő sorban látható a *Norm* érték, amivel azt határozzuk meg, hogy a tárgyat

alkotó háromszögek normálvektorai mekkora hatással legyenek a részecskék kiindulási sebességére, míg az *Ob* értékkel a Blender a tárgy helyzete alapján számítja ki a kezdősebességet. Itt a *Norm* értéket állítsuk 0,9-re, a *Rand* értéket pedig 0,2-re. A *Rand* változóval a részecskék kezdősebességét véletlenszerűen változtathatjuk. A *Tex* beállításával a tárgy felületi mintázata fogja befolyásolni a sebességértékeket. Ez alatt az érték alatt állíthatjuk be, hogy a mintázat milyen módon befolyásolja a kezdősebességeket. A három koordinátatengely mentén külön-külön beállítható, hogy a mintázat mekkora hatással legyen a részecskék sebességére, erre szolgál a X, Y és Z érték. Mellette látható a *Int* kapcsoló, amivel azt adjuk a Blender tudtára, hogy a mintázat képpontjainak világosság-értékét vegye figyelembe. A *RGB* kapcsoló használatával a mintázat színösszetevői szorzótényezőként szerepelnek majd a kezdősebesség meghatározása során, a *Grad* kapcsolóval pedig a világosságértékek változása befolyásolja a részecskék kezdősebességét.

Ezektől a beállítószervektől balra lévő értékekkel a megfelelő koordinátatengelyekkel párhuzamosan állandó értékű erőhatást adhatunk a részecskerendszerhez, azaz egy térbeli vektor összetevőit határozhatjuk meg. Itt állítsunk be (-1)-es értéket a Z tengely irányában, ami majd a gravitáció szimulálására lesz alkalmas.

Adjunk a jelenethez fényforrást, és a kamerát helyezzük el nagy távolságra a részecskerendszertől. Nagy távolság alatt azt kell érteni, hogy a látvány teljességének élvezetéhez a részecskéket kibocsátó tárgynak célszerű a kamera látóterén belül lennie. Később a kamerát természetesen a megfelelő helyre tehetjük, de kezdetben az egész részecskerendszert ajánlatos láthatóvá tenni.

Ezek után már csak a végeredmény kiszámítása maradt hátra. Az F10 billentyűvel váltsunk át a számítási beállításokat megjelenítő nézetre, és a *Anim* gomb oszlopának alján található *End* értéket állítsuk százra. A nézet bal szélén még találunk egy kapcsolót *DispView* felirattal. Ezt bekapcsolva a számítási eredmények közvetlenül a kamera látóterét megjelenítő nézetben lesznek láthatók. Kapcsoljunk a kamera nézetére, és kattintsunk a *Anim* gombra. Ekkor a szemünk előtt fog kibontakozni a végeredmény. Természetesen ezt akár rögtön a háttértárra is menthetjük, méghozzá úgy, hogy az alsó nézet bal oldalán a *pics* mezőbe beírjuk az állomány nevét, és az *Anim* gombtól jobbra, az az alatt található, alapértelmezésben *Targa* feliratú listáról kiválasztjuk valamelyik *AVI* típust. A CD-melékleten található animáció elkészítése során az *AVI JPEG* típust használtam, ezért ebben az esetben is ezt tudom javasolni. Mindezeket a beállításokat célszerű az animáció kiszámítása előtt elvégezni, ellenkező esetben a képsorozatot újra ki kell számíttatni a programmal.

Ebben a hónapban ennyi újdonság és lehetőség bemutatása véleményem szerint elegendő alapot adhat arra, hogy továbbra is mindenki kedvét lelje a program használatában. Az új billentyűkombinációk szokás szerint az összefoglaló *táblázatban* találhatóak meg.

Kellemes alkotást kívánva búcsúzóan.



Fábán Zoltán (dzooli@freemail.hu, dzooli@yahoo.com) 25 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.