A Blender kihasználása (2. rész)

Utolsó teendőként készítsük el játékunkat a Blenderben.

Mindenekelőtt fel szeretném hívni a figyelmet a CD-mellékleten található programra, amelynek segítségével a Blender 2.23-as változatában létrehozott objektumainkat szöveges formába menthetjük. Korábban már szóltam az UV koordináták mentésére készített programról, ennek továbbfejlesztését most olvasóink is használatba vehetik. A program alkalmas arra, hogy mentse a jelenetben található tárgyak pontjait, síklapjait, normálvektorait és természetesen a textúrakoordinátákat is. A forráskód elején megjegyzésben látható a kimeneti fájl formátuma.
A program a fájlokat a `defpath` változóban megadott könyvtárba menti. Használatához először be kell töltenünk a SHIFT-F11 billentyűk hatására előbukkanó szövegszerkesztőbe, majd az ALT-P billentyűkombinációval elindíthatjuk. Amennyiben sikeresen lefutott, a meghatározott könyvtárban megtaláljuk a mentett objektumokat, amelyeket későbbi felhasználás céljából tovább is alakíthatunk.

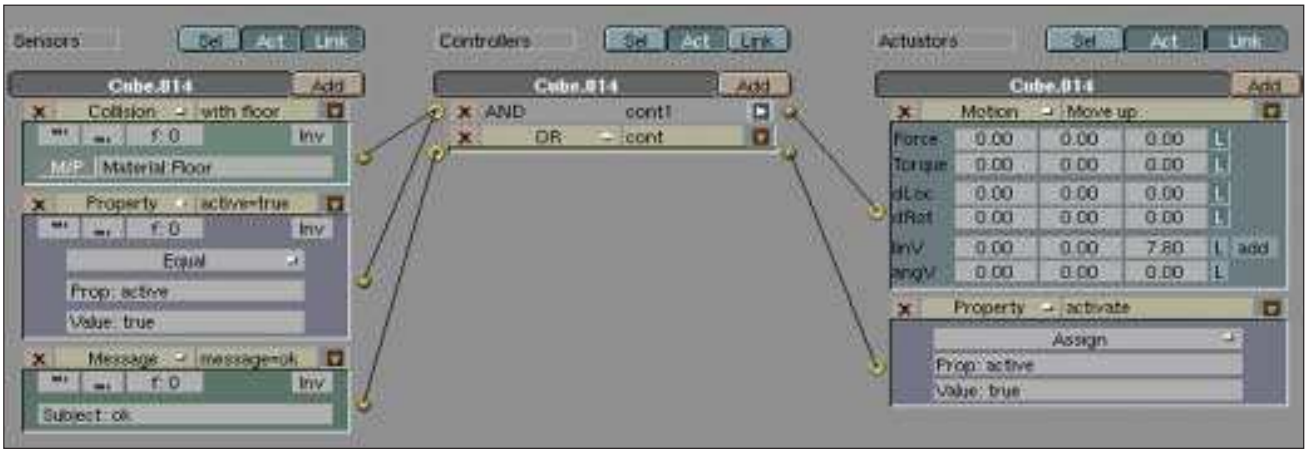
A játékterv

Ezek után kezdjük el a játék elkészítését. Egy-egy játék vagy bármilyen más program elkészítésének első lépése mindig a tervezés. Meghatározzuk, hogy mit kell tudnia a játéknak, hogyan kell működnie. Első egyszerű játékunk működése abból fog állni, hogy a játékos által irányítható figurát a pálya elejéről el kell juttatnunk a pálya túloldalán található kijárathoz, anélkül, hogy közben leesnénk a padlót képező síkról vagy valamelyik fel-le mozgó „kalapács” alá kerülnénk. A játékos a kurzormozgató billentyűkkel irányítható, és kapcsolatba kerülve az álló akadályok egyikével, azzal rugalmasan ütközik, majd arrébb pattan. A játékos dolgát nehezítik a pályán található fel-lemozgó kalapácsok, amikkel érintkezve a figura életét veszti, és a játék véget ér. Kiegészítésként még határozzunk meg annyit, hogy a játék kezdetét egy forgó, majd lefelé eső START GAME felirat jelzi, míg a végét egy GAME OVER felirat és egy YOU WIN felirat, amelyek forgás után újraindítják a játékot. Természetesen a ***Start*** felirat nem fogja újraindítani a játékot.
A fentiek alapján már tudjuk, hogy milyen tárgyakra lesz szükségünk. Készítsünk egy kockát, amit a hosszanti tengelye mentén méretezzünk át. Ez lesz majd a kalapács. Készítsünk egy újabb kockát, ezt azonban hagyjunk meg eredeti méretében, hogy a játék során megkülönböztethessük a kalapácsoktól. Ez a tárgy lesz – a sokszorosítás után – az álló akadály. Készítsünk két síklapot: az egyik lesz a pálya padlója, amin a figura mozoghat, a másik pedig azért fontos, hogy érzékelhessük azt, amikor a figura elhagyja a padlót és lefelé esve összeütközik ezzel a másik síklappal. Ezt a könnyebb elképzelhetőség érdekében tengernek vagy lávának nevezhetjük, így biztosan tudjuk, hogy káros lesz a játékfigura egészségére.
Szükségünk lesz még egy játékfigurára is. Az egyszerűség kedvéért most készítsünk egy gömböt, aminek az egyik vízszintes tengelye mentén húzzuk ki néhány pontját. Ez a pár kiálló pont lesz majd a figura „orra”, ebből tudja majd a játékos meghatározni a haladási irányát.

Az utolsó elkészítendő tárgy a kijáratot jelképezi. Ennél az olvasó képzelőerejére bízom a formai dolgokat, elegendő akár egy újabb kocka vagy gömb is, de lehet egészen bonyolult formát is alkotni. Az én példámban egy boltíves kaput készítettem. Elkészítettük tehát a tervet, megalkottuk a szereplőket, most már csak a működést kell vázolni, mielőtt a játék logikáját a Blender játékmotorjának a segítségével megvalósítanánk. Nagy vonalakban arról van szó, hogy amíg valamilyen felirat látható a képernyőn, addig a figura nem irányítható a billentyűzettel, és a tárgyak sem mozoghatnak, tehát az összes mozgó tárgynak a játékmotorban egy változóra lesz szüksége, amiből tudomást szerezhet a játék állapotáról. Amikor azonban semmilyen felirat nem szerepel, akkor a játék figyeli a billentyűzetleütéseket, így mozgathatjuk a figurát. Három tárggyal ütközhetünk. A padló alatti „tengerrel”, a „kalapácsokkal” és a kijáratot jelképező kapuval. Amikor a figura a kapuval ütközik, megjelenik a YOU WIN felirat, majd elindul egy számláló, amíg a felirat felfelé mozog, majd egyhelyben megáll. A meghatározott számérték elérése után a játék elölről kezdődik. Amikor a figura egy „kalapáccsal” vagy a „tengerrel” ütközik, szinte ugyanez a folyamat játszódik le, csak a GAME OVER felirat jelenik meg. A játék kezdetén megjelenik a START GAME felirat, meghatározott ideig forog, majd a másik számláló elindításával lefelé mozog. A figura és a kalapácsok mozgása a játék állapotát jelző változóhoz van kötve, csak annak igaz értéke mellett kell figyelembe venni.
Azt is tudjuk, hogy a kalapácsoknak pattogniuk kell, ez azonban az előző részek figyelmes elolvasása után már nem okozhat gondot. Tehát hozzuk létre a tárgyat, adjunk meg dinamikus anyagtulajdonságokat, az ütközés esetére pedig egy mozgatóerőt, amellyel az eredeti magasságába pattanhat vissza. Hogy ne kelljen minden ilyen veszélyes tárgyat újból létrehoznunk, a SHIFT-D billentyűkkel készítsünk róla másolatokat. A másolatokat mozgassuk el a helyükről, majd az utoljára létrehozottnak adjunk más tulajdonságokat, és erről szintén készítsünk másolatokat. Erre a lépésre azért volt szükség, hogy a játékos dolga ne legyen olyan egyszerű: más tulajdonságok esetén ugyanis más sebességgel fog pattogni az akadály.

```
Végtelen ciklus
Ha (aktív és ütközik(padló)) akkor
      erőhatás_felfelé
Ha (üzenet = "ok") akkor
      aktív := hamis
Ciklus vége
```

Miután minden mozgó akadályt elkészítettünk, helyezzük el őket a pályán. A mozgásukat vezérlő logika az *1. képen* látható. A jobb érthetőség kedvéért az *1. listán* egyszerű nyelven megfogalmazva láthatjuk a vezérlés logikáját.
Most következzen a játékfigura vezérlésének a megvalósítása. Szükségünk lesz egy logikai típusú tulajdonságra, amit



1. kép A kalapács logikája

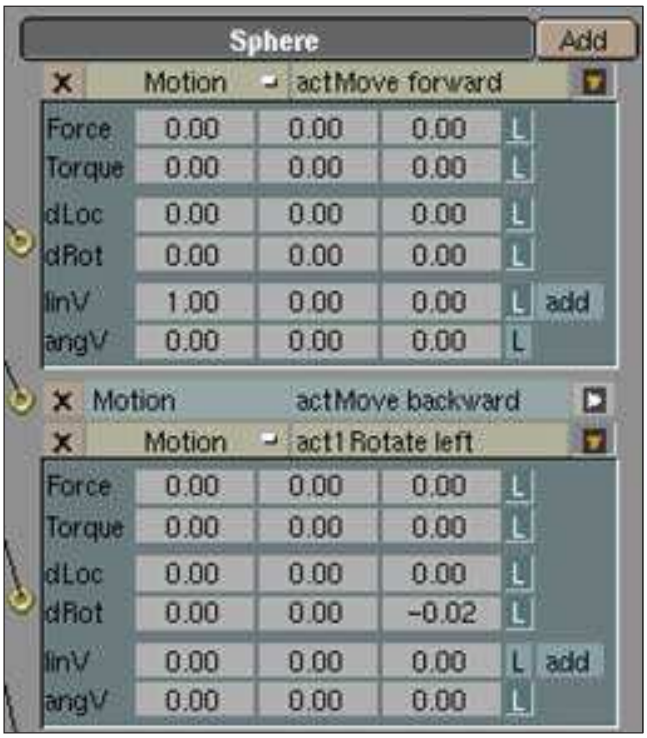
nevezzünk ***active***-nek. Ennek igaz értéke jelzi azt, hogy a játék fut és a figura irányítható. Továbbá a tulajdonságot lekérdező ***Propety*** érzékelőre is szükségünk van. Ezek mellett természetesen négy billentyűlenyomás-érzékelőt is létre kell hoznunk, mindegyik kurzormozgató gombhoz egyet-egyet. Ezeket a ***Keyboard*** típust választva készíthetjük el. Ne felejtsük el a megfelelő billentyűket beállítani. Az ütközések kezeléséhez készítsünk még három ütközésérzékelőt is (***Collision***), ezeknél anyagtulajdonságként a kapu, a kalapácsok és a „tenger” anyagát adjuk meg. A mozgás megvalósításához természetesen a mozgás tulajdonságaira ható ***Motion*** hatást kell létrehoznunk, szám szerint négyet. A gyorsításhoz és a lassításhoz (amelyeket a fel és le nyilakhoz rendelünk hozzá) egyet-egyet; ezeknek a `LinV` értéke fontos, itt adjunk meg 1-et, valamint -1-et az első koordináta szerint. A jobbra és balra nyilakhoz a Z tengely körüli elfordulást kell meghatároznunk, tetszés szerinti értékkel, de ne felejtsük el bekapcsolni a `Loc` kapcsolókat. Ezek a beállítások a *2. képen* láthatók.

Tudjuk, hogy a tárgynak csak akkor szabad mozognia, ha a játék futása megengedett, vagyis ha az ***active*** tulajdonság igaz. Valahogyan ezt a tulajdonságot is be kell állítanunk, tehát a kezdőértékét állítsuk hamisra, és hozzunk létre egy új érzékelőt, ***Message*** típussal. A ***Subj:*** mezőben adjuk meg a `start` szócskát, mert a későbbiekben ez az üzenet fogja jelezni a játék indulását. Az üzenetet majd a START GAME szöveg küli a jelenet minden aktív szereplőjének. Ehhez az üzenethez egy tulajdonságbeállító hatásnak kell tartoznia, ami az ***active*** értékét igazra fogja állítani. Most következhet az ütközések kezelése. Amikor a figura a kalapáccsal vagy a „tenger”-rel ütközik, egy üzenetet kell küldenünk a játék végét jelző szövegnek. Ez az üzenet legyen mondjuk „gover”, vagyis hozzunk létre egy üzenetet (***Message***) és a ***Subj:*** mezőbe írjuk be a „gover” szót.

Hasonló módon oldjuk meg a kapuval való ütközést is. Annyi különbség azért van, hogy az üzenetet nem a kapunak küldjük, hanem a YOU WIN feliratnak, és értelemszerűen más lesz az üzenet tárgysorában szereplő szöveg is.

Inudhat a játék

A létrehozott elemeket még össze kell kötni egymással. A billentyűzeteseményeket `AND` kapcsolatba kell hozni az ***active*** tulajdonságot érzékelő elemmel, így biztosíthatjuk, hogy csak a megfelelő időben tudjuk mozgatni a figurát. A kezdetet jelző érzékelő kimenetét egy `OR` logikai kapcsolaton keresztül kapcsoljuk a tulajdonság beállításához. Az ütközéseket csak a játék futása során kell figyelembe vennünk. Itt `OR` logikai kapcsolatot



2. kép Gyorsítás és fordulás

használhatunk. Következhet tehát a kezdőüzenet megvalósítása. Szükségünk lesz egy ***Timer*** tulajdonságra, ennek a kezdőértékét állítsuk nullára. Két értékvizsgálatot kell elvégeznünk a vezérlés során: az egyik szerint, amikor az időzítő értéke 1 és 1,45 között helyezkedik el, a szöveget a kamera síkjával párhuzamos tengely mentén egy kicsit el kell fordítanunk. Itt szándékosan nem adtam meg, hogy mennyi legyen az a „kicsi”, célszerű kikísérletezni, azt is figyelembe véve, hogy hányszor szeretnénk körbefordítani a szöveget, mielőtt lefelé mozogva kúszna a kamera látóteréből. Tehát egy ***Property*** érzékelőnek a típusát állítsuk ***Interval***-ra, és adjunk meg két értéket. Ehhez egy `OR` kapcsolattal kapcsoljunk hozzá

P	A játék indítása
ESC	A játék leállítása
Shift-F11	Parancsállomány-szerkesztő

```
Idő:=100
Végtelen ciklus
Ha ( Idő>1 és Idő<20) akkor
      szöveg_mozog_fel
Ha ( Idő > 40 ) akkor
      játék_újraindul
Növel( Idő )
Ha ( üzenet("win") ) akkor
      Idő:=0
Ciklus vége
```

egy ***Motion*** hatást, így a megfelelő értékek beállítása után a szöveg már képes lesz a kezdeti forgásra. A lefelé haladáshoz szintén az értéktartományt kell vizsgálnunk; hozzunk létre egy másik ugyanilyen érzékelőt, majd kapcsoljuk hozzá egy újabb ***Motion*** hatáshoz és egy ***Message*** hatáshoz is. A mozgás mértékét szintén tapasztalatunk alapján tudjuk majd megállapítani, de ne felejtsük el ennek a második értékvizsgálatnak az alsó határát nagyobbra állítani, mint amekkora az elsőként beállítottnak a felső határa volt. Amennyiben ezt elmulasztjuk megtenni, a szöveg a forgással egyidőben fog mozogni, ezt viszont most nem szeretnénk. Az üzenet tárgya legyen „start”, ezzel jelezzük az aktív tevékenységre képes tárgyaknak, hogy a játék elkezdődött.

Az eddigieket figyelemmel kísérve odáig jutottunk el, hogy a játék kezdetén van egy forgó, majd leeső START GAME felirat, s ezután irányíthatjuk a figurát. A kalapácsok fel-le mozognak, és amikor a figura alájuk kerül vagy leesik a pályáról, üzenetet küldünk valamelyik másik tárgynak. Amikor a figura a kapunak ütközik, a játékmotor felhasználásával szintén üzenetet küldünk. Mint látjuk, már nem maradt sok dolgunk. Az ütközésekkor keletkező üzeneteket a megfelelő helyen fel kell dolgozni.

A „win” üzenetet a YOU WIN feliratnak kell értelmeznie. Az üzenet hatására fel kell bukkannia, majd rövid várakozás után újra kell indítania a játékot. A felbukkanást könnyen megoldhatjuk úgy, hogy a játék kezdetén a szöveget a pálya alatt helyezzük el, és az üzenet hatására egy időzítő értékét változtatjuk meg. Ezután, amíg az időzítő egy bizonyos tartományon belül tartózkodik, a szöveg felfelé mozog; majd egy másik értéktartományba belépve a játék elölről kezdődik. A dolgok gyakorlati oldalát nézve: hozzunk létre egy ***Timer*** típusú tulajdonságot a szöveg számára. Ennek a kezdőértékét állítsuk mondjuk 100-ra, így elkerülhetjük, hogy már a játék kezdetekor felvegye azokat az értékeket, amikre csak később lesz szükség. Hozzunk létre két értéktartomány-érzékelőt ***Property – Interval***, és a ***Prop*** mezőben adjuk meg az időzítő azonosítóját. Az első érzékelési tartománya legyen például 1–20, ez fogja vezérelni a mozgást, és egy OR kapcsolaton keresztül kapcsoljunk is hozzá egy mozgásra vonatkozó hatáselemet. A mozgás jellemzői ugyanazok lehetnek, mint az előző felirat esetében.

A másik érzékelő alsó határát állítsuk negyvenre, a felső határát pedig 99-re. Talán érthető, hogyha itt a felső határt 100-nál nagyobbra állítottuk volna, már a játék kezdetén véget érne a működése, mert az időzítő kezdőértékét 100-ra állítottuk. Az első vizsgálat felső határa és a második vizsgálat alsó határa közötti különbség (esetünkben ez az érték 20) adja meg azt az időt, amíg a felirat nem forog, és még a játék sem indul újra. Itt figyelni kell arra, hogy a forgás befejezésekor a felirat a megfelelő elfordulással jelenjen meg.

A játék újraindításához egy ***Scene*** típusú hatás kell az érzékelőhöz kapcsolni, majd ki kell választani a legördülő listából a ***Restart*** tételt. Most már minden rendben is volna, ha a felirat valamilyen módon tudomást szerezne arról, hogy mikor is kellene elkezdenie a mozgást.

A megoldás az lesz, hogy egy ***Message*** érzékelőnek beállítjuk a tárgy mezőjében a „win” szócskát (amit a figura–kapu-ütközéskor majd a játékfigura küld a jelenetben résztvevőknek), és egy OR kapcsolat segítségével egy olyan hatást kapcsolunk hozzá, ami valamilyen tulajdonságot állít be. Ennek a tárgynak az egyetlen tulajdonsága az időzítő értéke, vagyis ezt nullára kell beállítanunk. Ezzel elérjük, hogy a kapuhoz érve az időzítő nulla értéket kap, majd folytatja természetes működését, vagyis az értéke növekedni kezd. Egy és húsz közötti értékek között a szöveg felfelé mozog, ezután 20–40-ig nem történik semmi. A negyvenes értéket elérve a játék a ***Scene*** hatás működése miatt elölről kezdődik. A fentiek megértését segíti a 2. *lista*.

A játékos megsemmisülésekor ugyanezt a vezérlést kell megvalósítani, csak ebben az esetben a GAME OVER felirathoz kell hozzárendelni a megfelelő elemeket, és az üzenet tárgysorában a „win” szöveg helyett a „gover” szövegnek kell szerepelnie, hiszen – mint korábban már beállítottuk – a kalapáccsal vagy a „tenger”-rel ütköző figura ezt az üzenetet küldi a jelenet résztvevőinek.

Nos, ennyi lenne egy egyszerű játék elkészítése a Blender játékmotorjának a felhasználásával. A játékot a P billentyűvel indíthatjuk el, és máris észrevehetjük eddigi munkánk első szépséghibáját. Látható, hogy minden fehér színben és árnyékolás nélkül jelenik meg. Ha a megjelenítést a D billentyűvel előhívható menüből választva árnyékoltra változtatjuk, és így indítjuk el a játékot, látjuk, hogy ettől sem lett jobb a helyzet. Nos, ha figyelemmel kísérték a sorozat korábbi részeit, nem fog gondot okozni a felületi mintázatok elkészítése. Szükség is lesz e munka elvégzésére, ugyanis a Blender csak ilyen mintázatokkal ellátott tárgyakat jelenít meg látványosan a játékmotor elindítása után. Ezt a lépést sajnos nem célszerű a teljes vezérlés és a környezet kialakítása után elvégezni, mert így elveszítjük azt a lehetőséget, hogy a másolatként keletkezett tárgyaknak már megvan a mintázata és minden egyéb tulajdonsága is. Ha most szeretnénk elkészíteni a mintázatokat, akkor azokat minden tárgyon külön kellene beállítanunk. Ez véleményem szerint felesleges lenne, létezik más megoldás is. Ahhoz, hogy a mostani munkánk se vesszen kárba, nem kell teljesen elölről kezdeni mindent, elég, ha minden másolatból csak egyet hagyunk (egy kalapácsot és egy álló akadályt), majd ezeket mintázatokkal ellátva megismételjük a másolatok készítését és elhelyezését.

A mintázatok

Természetesen a játék még nincs befejezve, de a célom nem is ez volt, hanem az, hogy bemutassam egy jól használható modellező és animációs program képességeit, olyan eszközöket és módszereket adva az érdeklődők kezébe, amelyek nem csupán szórakoztató és unaloműző tevékenységre alkalmasak, hanem az egyéni önkifejezésre is.

A továbbiakra nézve sok-sok jó ötletet és az alkotás örömét kívánva búcsúzom.

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