Inverz kinematika a Blenderben

Ismét eljött az ideje, hogy új tudásra tegyünk szert a Blender képességeinek megismerése során.

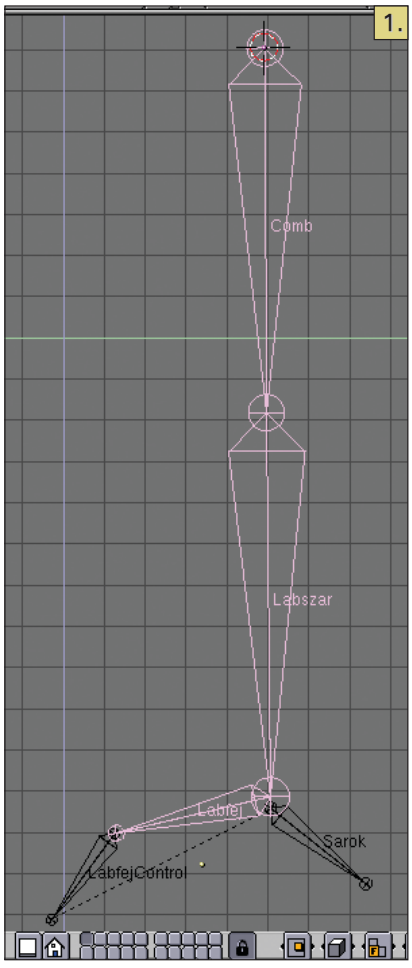
Remélem, kedves olvasóim, mára már sikerült kellőképpen elmélyedni a korábban elsajátított ismeretekben, így a továbbiakban nem fog gondot okozni a program használata és különféle lehetőségeinek megértése.
Ebben a hónapban azt szeretném röviden bemutatni, hogy karaktereinkhez hogyan készíthetünk inverz kinematika által mozgatott csontvázakat. Kezdetben egy egyszerű módszert szeretnék ajánlani a karaktermodellezés mikéntjével kapcsolatban. Egy csomó felesleges munkától kímélhetjük meg magunkat, ha betartjuk az alábbi sorrendet.

A terv

Először gondoljuk át, hogyan fog kinézni a karakter, majd készítsünk minél részletesebb rajzokat róla. Ezeket később a modellezés során is felhasználhatjuk, hiszen a pontos modellezéshez elengedhetetlenek. Természetesen a mintázatokat is könnyebben elkészíthetjük, ha egyszer már megrajzoltuk őket. Készítsünk elöl-, oldal- és felülnézeti képeket, és a modellezőprogramban használjuk őket háttérképként és segítségként egyaránt. A rajzok mellett legalább elképzelés szintjén szükségesnek tartom a háromszögek elhelyezkedésének átgondolását is. A következő lépésben tervezzük meg, hogy milyen mozgásokat kell végeznie a figurának, és gondoljuk át a csontváz szerkezetét. Modellezés során a csontváz létrehozásával kezdjük, erre könnyebben felépíthetjük majd a figurát. Ezután következhet a modell elkészítése, majd a mintázatokat képolvasóval és rajzprogrammal elkészítjük és finomítjuk. Utolsó előtti lépésként a modell mintázatának elhelyezése és pontosítása következzen, ezek után a mozdulatsorokat modellezzük a csontváz mozgatásával.
A fentiek betartásával könnyebben és rövidebb idő alatt tudunk majd mozgó figurákat modellezni, és szemet gyönyörködtető animációk vagy játékok készítéséhez használhatjuk fel őket.
Most tehát ismerkedjünk meg az inverz kinematika létrehozásának lehetőségeivel!

Mozgás!

Mint azt a korábbi részekben említettem, az inverz kinematika teszi lehetővé, hogy amikor valamilyen tárgyat mozgatunk, annak mozgása ne csak önmagában létező legyen, hanem a mozgás során az összekapcsolt tárgyak egymásra hatása is megjelenjen. Egyszerű példaként az emberi kar mozgása hozható fel, hiszen nem helyezhetem tetszőleges helyre a kézfejemet anélkül, hogy az alkar és a felkar is ne mozogna, és a kézfejhez kapcsolódó testrészek hosszúsága is korlátozza a kézfej mozgását.
Úgy gondolom, ennyi bevezető után elkezdhetjük a gyakorlati munkát. Egy három ízből álló csontváz mozgását tanuljuk meg, ennek alapján a későbbiekben bonyolultabb vázakat is képesek leszünk mozgásra bírni. Oldalnézetben hozzunk létre egy három részből álló vázat a ***Főmenü/Add/Armature*** menüpontok kiválasztásával. Az egyes részeket nevezzük el fentről lefelé, a „Comb”, „Lábszár” és a „Lábfej” megnevezéseket használva. Ehhez ki kell jelölni a vázat, majd a TAB billentyű alkalmazása után minden részt is az A billentyűvel. Ekkor az F9 billentyűvel váltsunk át a szerkesztőnézetbe, majd a nézet közepe felé látható „Bone” neveket változtassuk a fentieknek megfelelőkre. Ezt csupán azért kellett megtenni, hogy a kedves olvasók hozzászokjanak a nevek használatához, hiszen amikor emberi csontvázat készítünk, könnyebben eligazodunk az ismerős nevek között, mint a számozott csontok között. Természetesen a továbbiak megértése is könnyebben megy majd, ha például a leírás szerinti „Comb” megtalálásához nem kell minden esetben a szerkesztőnézetben keresgélni a csontok között. Miután elneveztük a csontokat, még egy két részből álló vázat kell létrehozni, amivel majd befolyásolhatjuk a mozgást. A szerkesztőmódban jelöljük ki a boka helyén lévő kör alakú csatlakozási pontot, majd a SHIFT-S billentyű hatására megjelenő menüből válasszuk a legalsó pontot. Így a térbeli kurzor pontosan a boka helyére kerül. Szüntessük meg a kijelölést, és az előbbi módon hozzunk létre egy új vázat. Mivel ennek az irányító váznak a részei nem csatlakoznak egymáshoz, újra szüntessük meg a kijelölést, és a fő vázon a lábfej végét kijelölve helyezzük el ide a kurzort (a SHIFT-S ismételt alkalmazásával). Szüntessük meg ismét a kijelölést, majd a vezérlő vázat kijelölve váltsunk szerkesztőmódba. Itt következik a másik vezérlőrész létrehozása, vagyis a főmenüből ismételten válasszuk ki az ***Armature*** pontot. Ezek után a létrehozott csontoknak adjuk a „Sarok”, illetve a „LábfejControl” nevet. Amennyiben mindent jól csináltunk, a továbbiakban



1. kép Kiindulás az inverz kinematikához

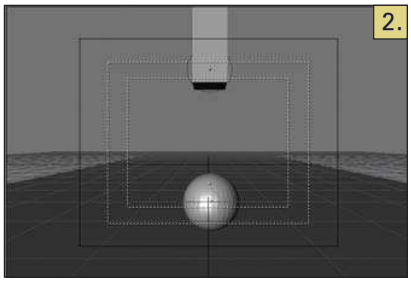
az *első* képen látható szerkezettel dolgozhatunk. Megjegyzem, hogy a képen a mozgatandó váz van kijelölve, tehát az látható rózsaszínnel jelölve, a fenti lépések után azonban a Blenderben még nem látható a „LábfejControl”-t és a „Sarok” vastagabbik végét összekötő szaggatott vonal. Ez a vonal jelenti azt, hogy a két csont alárendelt viszonyban áll egymással, és a helyes mozgáshoz ezt is be kell állítani. Ezt a két csontot a szerkesztőmódban kijelölve a szerkesztő nézetben megjelenik a csontok neve és az is, hogy milyen viszonyban vannak egymással. Ezt az információt a név mellett látható ***child of*** mező tartalmazza, és nekünk kell most beállítani a „Sarok”-nak ezt a tulajdonságát. Tehát a „Sarok” sorában válasszuk ki a legördülő listából a „LábfejControl” csontot. Így a létrehozott szerkezet már valóban megfelel az első képen láthatónak.

Inverz kinematikai rendszer

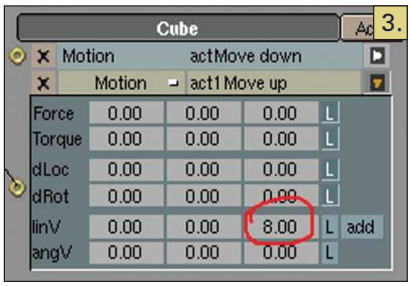
Ezek után nem sok teendőnk van, csupán a Blenderrel kell tudatni, hogy ezek a részek tulajdonképpen egy inverz kinematikai rendszert alkotnak. Itt lesz szükségünk az előző részben olvasottakra, nevezetesen a ***helyzetmód*** alkalmazására. Először azonban az esetleges elfordulásokat és az egyéb változtatásokat mindkét vázrészen meg kell szüntetni. Ezt az Alt-R billentyűkkel érhetjük el, de ne lepődjünk meg, ha a vázak helyzete megváltozik, hiszen így alaphelyzetbe kerülnek, ami megkönnyíti további munkánkat. Végezzük el ezt a műveletet mindkét vázon. Ezután jelöljük ki a három részből álló csontvázat, majd a Ctrl-Tab billentyűkkel váltsunk ***helyzetmódba***. Néhány ablakkezelő ezt a billentyűkombinációt az alkalmazások közötti váltásra tartja fenn, ezért helyette használhatjuk a nézet alján lévő gombok közül azt, amelyiken egy sárga fejecske látható. Ez a rétegek megjelenítéséért felelős gombok után a tizenkettedik.
Amikor ebben a módban dolgozunk, a kijelölt csontváz kék színnel jelenik meg. A sikeres váltás után jelöljük ki a „Lábfej”-et, és a szerkesztőnézet bekapcsolása mellett látható, láncszemeket ábrázoló gomb segítségével váltsunk át a korlátozásokat és kapcsolatokat meghatározó nézetre. Itt kell majd megadnunk, hogy melyik rész milyen kapcsolatban van a többiekkel. Az ***Add*** gombbal hozzunk létre egy új elemet, és a megjelenő vezérlők között keressük meg a ***Constraint type*** mezőt. Ez a kis x mellett található, és a listából ki kell

Alt-R	Elfordulás törlése
F8	Váltás játéknézetre
P	Játék indítása
Esc	Játék leállítása

választanunk az `IK Solver` típust. Ennek hatására a típus alatt megjelenik egy `OB` szerkesztőmező, ahová be kell írni azt az objektumot, amellyel vezérelni fogjuk a csontváz mozgását. Ebben az esetben nem neveztük át a két csontból álló vázat, így annak neve ***Armature.001*** maradt. Ezt kell a billentyűzet segítségével beírnunk az `OB` mezőbe. Ahogyan ezzel elkészültünk, megjelenik egy újabb szerkesztőmező, ahová a „Sarok” nevet kell beírni, hiszen ez a csont fogja vezérelni a teljes csontváz mozgását. Ezzel készen is vagyunk. Szüntessük meg a kijelölést, és a két részből álló váz mozgatásával máris egy működő rendszert láthatunk.
Felmerülhet a kérdés, hogy ha ez ilyen egyszerű, akkor miért volt szükség a „LábfejControl” vázrész létrehozására. Nos, ezzel tudjuk majd a lábfej elfordulását szabályozni. Ennek eléréséhez azonban újra az előbbi korlátozó tényezőkhöz kell újakat adnunk. Jelöljük ki megint a fő vázat, majd helyzetmódba váltva a „Lábfej”-et is. Adjunk újabb korlátozó elemet az ***Add*** gombbal a vázrészhez, ezúttal azonban az alapértelmezett ***Track to*** típusú elemen a típust hagyjuk változatlanul. Az `OB` ismét az ***Armature.001*** legyen, azonban az elemhez tartozó csontként most a „LábfejContol”-t kell beállítani. Az eredményeket megszemlélve látható, hogy a vázrendszer a „Sarok” mozgatásával most megfelelően mozgatható, és a két részből álló váz forgatásával a lábfej külön is képes forogni. Ha jobban megfigyeljük, akkor azt is észrevehetjük, hogy a mozgás nem tökéletes. Néhány helyzetben az egész láb kifordul, nem tudjuk például vízszintesen előrenyújtani. Ezen további korlátozó elemek létrehozásával könnyen segíthetünk. Szintén a fő vázon kell dolgoznunk, de első lépésben a „Comb”-ot jelöljük ki. Adjunk egy újabb ***Track to*** elemet a vázrészhez, ennek célját állítsuk az ***Armature.001*** objektum „LábfejControl” csontjára. A következő lépés legyen a „Lábszár” elemhez egy újabb korlátozó tényező hozzáadása, célja szintén az ***Armature.001*** váz, de itt a követendő csont a „Sarok” lesz. Ezeket a lépéseket megtéve már minden pozícióban helyesen fog állni



2. kép Kiindulás a játékhoz

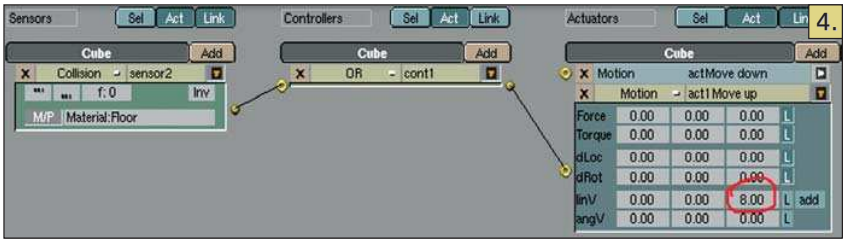


3. kép A kezdősebesség beállítása

a háromízű váz. Itt kell megemlítenem, hogy a ***Track to*** elemet más esetekben is eredményesen használhatjuk, például amikor a kamerának követnie kell egy objektum mozgását – alkalmazzuk bátran ezt az elemet. A ***Copy Location*** és a ***Copy Rotation*** elemek használata magától értetődő, így ezekkel most nem foglalkozom részletesen.

Játék megalkotása

Úgy gondolom, hogy mostanra már eleget tudunk a Blenderről ahhoz, hogy elkészítsük első egyszerű játékunkat. A jobb érthetőségért célszerű tisztáznunk néhány alapvető dolgot.
A Blenderben a játékmotorban négyféle elemtípust találhatunk. Az egyik a tulajdonság, amely egy-egy tárgyunk sajátja, és típusa a különféle programozási nyelvekben megszokottaknak megfelelően lehet egész, valós, logikai és szöveg. Itt lehetőségünk nyílik egy úgynevezett ***Timer*** típusú tulajdonság használatára is, ennek kezdőértéket adhatunk, majd az értéke önműködően növekszik. A második típus az érzékelő, amellyel különféle események bekövetkezésekor indíthatunk el valamilyen folyamatot. Ennek típusa is többféle lehet, itt azonban csak a ***Keyboard***, a ***Always*** és a ***Collision*** típusú érzékelőket fogjuk használni – ezekkel részletesen foglalkozom. A ***Keyboard*** alkalmas különféle billentyűzettől érkező események figyelésére, megadható, hogy melyik billentyű lenyomására legyen érzékeny, milyen módosítóbillentyűkkel



4. kép A pattogás logikája

együtt figyelje a billentyű lenyomását, és hogy amikor az adott billentyűt lenyomtuk, akkor a tárgyunk milyen tulajdonsága változzon meg. Az ***Always*** típusa érzékelő, tehát a megadott gyakorisággal jelzéseket továbbít a kimenetén. A ***Collision*** alkalmas arra, hogy egy megadott tulajdonsággal vagy anyagtípussal rendelkező tárggyal való ütközést érzékeljen.

Az eddigieket próbáljuk megvalósítani a gyakorlatban is. Kiindulásként hozzunk létre egy gömböt, egy síklapot és egy kockát, majd ezeket helyezzük el a *2. képen* látható elrendezésben.

A játék célja az lesz, hogy egy akadálypályán keresztül kijussunk a gömbbel a kijáraton. A kijárat a pálya másik végén lesz.

Kezdetben adjunk anyagot a síklapnak, és ezt az anyagot nevezzük „Floor”-nak. Ezután a kockát kijelölve az F8 billentyűvel kapcsoljunk játéknézetbe. Itt állíthatjuk be a játék működését meghatározó tulajdonságokat, érzékelőket és egyéb, a működést befolyásoló elemeket, amelyekről a későbbiekben bővebben szólok. Most a kocka kijelölése után kapcsoljuk be a nézet bal felső sarkánál található `Actor` kapcsolót, ezzel tudatjuk a Blenderrel, hogy ennek a tárgynak a tulajdonságait a játék futtatása során figyelembe kell vennie. Kapcsoljuk még be a megjelenő `Dynamic` kapcsolót is. Ezzel a fizikai törvények is hatni fognak a tárgyra, ami most elsősorban a gravitáció miatt lesz fontos, hiszen ez lesz majd az egyik akadály, a labdának ami a pálya síkja felett pattogni fog. Ha most a P billentyűvel elindítjuk a játékot, akkor már láthatjuk is a mozgást, vagyis azt, hogy a tárgy elkezd lefelé esni, és amikor érintkezik a síklappal, akkor megáll. Ez már fél siker, célunk ugyanis a folyamatos pattogás. Ezt úgy tudjuk megvalósítani, hogy egy érzékelőt helyezünk a tárgyra, ami a síklap elérésekor jelez, és ekkor a kockának új irányt adunk.

Tehát a következő lépésben a nézet közepe felé a ***Sensors*** felirat alatt találunk egy gombot, amivel új érzékelőt adhatunk a tárgyhoz. Az ***Add*** gomb alkalmazásával hozzunk létre egy új elemet, ennek típusát pedig a megjelenő legördülő listából kiválasztva állítsuk ***Collision***-ra. Itt található még egy ***M/P*** kapcsoló is, amivel választhatunk, hogy az adott anyaggal vagy tulajdonsággal rendelkező tárggyal való ütközéskor jelezzen-e meg az érzékelő. Ezt a kapcsolót is kapcsoljuk be, majd a mellette lévő mezőbe írjuk be a „Floor” szócskát. Ezzel elértük, hogy az érzékelő olyan tárgyakkal való ütközéskor jelezzen, amiknek az anyaga „Floor” nevű. Ilyen tárgy jelenleg egy szerepel a jelenetben, a síklap. Az érzékelő beállítása után már csak azt kell megadnuk, hogy mi történjen a jelzés hatására. A nézet jobb oldalán találhatók az események eredményeként létrejövő hatások. Itt is alkalmazzuk az ***Add***-ot, és az alapértelmezett hatás megjelenése után állítsuk be annak jellemzőuit. Amikor a kocka eléri a síklapot, arra van szükség, hogy újra felfelé induljon el. Úgy biztosítunk számára kezdősebességet, hogy a `linV` érték Z irányú összetevőjének pozitív értéket adunk meg. Ez látható a *3. képen*, a szükséges változtatás pirossal van jelölve. Én a példámban 8-as kezdősebességet határoztam meg, ami arra elegendő, hogy a kocka addig a magasságig pattanjon fel, ahonnan elindult.

Rugalmasság

Ideje, hogy egy, az eddigiek során kissé elhanyagolt anyagtulajdonsággal is foglalkozzam, a dinamikai tulajdonságokkal. Az anyagszerkesztőben a színek meghatározása mellett találhatunk egy ***Dyn*** feliratú gombot, ezt bekapcsolva megadhatjuk az adott anyag rugalmasságát, tapadási együtthatóját és azt, hogy mennyi perdületet szerezzen az ütközés után. Ezek használatához azonban a játéknézetben is be kell állítani a ***Actor*** gomb alatt található `Do Fh` kapcsolót.

Miután létrehoztuk az érzékelőket, és meghatároztuk, hogy a jelzés hatására a tárgynak milyen tevékenységet kell végeznie, már csak a kettő közötti kapcsolatot kell meghatározni. Erre szolgálnak az érzékelők és a hatások közötti területen lévő vezérlők. Ezeknek négy fajtája létezik, az `AND` ***és*** kapcsolatot jelent a kapcsolódó bemeneti jelzések között, az `OR` ***vagy*** kapcsolatot, míg a fennmaradó kettő működése általam jelenleg nem ismert. Szerencsére ez nem jelent gondot, hiszen a meglévő érzékelők és kapcsolatok elegendőek a játék elkészítéséhez. Válasszuk ki tehát a ***vagy*** kapcsolatot, és kössük össze a kimeneteket és bemeneteket. Ezek a sárga színű körök és a telt körök az egyes elemek mellett. Telt körrel jelölve láthatjuk a kimeneteket, amelyeket egyszerű kattintással és húzással az üres körrel jelölt bemenethez köthetünk. Végső lépésként kössük az értékelő kimenetét egy ***vagy*** kapcsolat bemenetéhez, majd ennek kimenetét a mozgatást megvalósító hatás bemenetéhez. A *4. képen* látható a helyesen kialakított vezérlőrendszer.

Ebben a részben eddig jutottunk, az összefoglaló táblázatban mindenki megtalálja a tanult billentyűkombinációkat; a következő hónapban pedig a billentyűzetkezeléshez használható érzékelőket mutatom be majd, és folytatjuk a játék elkészítését.

Fábián Zoltán
(dzooli@freemail.hu)
Programozóként dolgozik. Szabadidejében szívesen kirándul, és szeret rajzolni, érdekli a 3D grafika.