

Az ObjectBender használata

Szükséges előzetes tudás:

- objektumkészítés

Készítette: Phonteus Nevolius *Michelle* útmutatója¹ alapján

Legutóbbi módosítás: 2021. július 31.

Készült: 2010. augusztus 14.



1. Bevezető

Az ObjectBenderrel össze lehet fűzni egy objektumot több kisebb szegmensből és meghajlítani, ívelni azt. Ez jól jön, ha például sín, fal vagy töltés objektumot készítünk. A szegmensek .B3D vagy .CSV formátumúak lehetnek. Ezeket a fejlesztőnek el kell tudnia készíteni, így értenie kell az objektumkészítéshez.

Az ObjectBender használható grafikus és parancssoros felületen is. Parancssoros kapcsolók nélkül indítva a program grafikus felületéhez jutunk. A grafikus felület hasznos a program megismeréséhez, de hosszú távon célszerű a parancssoros felületet használni. A parancssoros felületen használható kapcsolókat alább részletezzük.

Vegyük figyelembe, hogy az ObjectBender UTF8-as karakterkódolásúként értelmezi a fájlokat, ha csak a byte sorrend jel nem mutat más unicode karakterkódolást. A nem unicode karakterkódolások nem támogatottak.

Az ObjectBendert megtaláljuk az OpenBVE telepítési könyvtárában, az *ObjectBender.exe* fájlt fogjuk használni.

2. A parancssoros felület kapcsolói

Kapcsoló	Jelentése
/?	Kilistázza az alkalmazható parancssori kapcsolókat.
BemenetiFájl	A bemeneti fájl elérési útja. Lehet .B3D vagy .CSV.
KimenetiFájl	A kimeneti fájl elérési útja. Ugyanolyan formátumú lesz, mint a bemeneti fájl.
/n=Szegmensek	A szegmensek száma.
/s=SzegmensHossz	Az egyes szegmensek hossza méterben.
/b=BlokkHossz	Blokkhossz. Ha nullát adunk meg, az objektumot nem forgatja el a program a sínként való használathoz.
/r=Sugár	A sugár méterben. Negatív értékek balra, pozitív értékek jobbra hajlítják az objektumot. Nulla esetén nem történik hajlítás.
/g=Nyomtáv	Nyomtáv milliméterben. Csak túlemelés használata esetén kell megadni.
/u=KezdőTúlemelés	A túlemelés értéke az objektum elején.
/v=VégsőTúlemelés	A túlemelés értéke az objektum végén.
/a	Beleírja a változtatásokat a kimeneti fájlba ahelyett, hogy felülírná azt.

Példa:

```
| ObjectBender input.csv output.csv /n=25 /s=1 /b=25 /r=600
```

3. Az ObjectBender működése

Az ObjectBender másolatokat készít a kívánt objektum általunk elkészített szegmenséről a Z tengely mentén. Minden szegmens az általunk megadott szegmenshossz értékével távolabbra kerül a Z tengelyen. Az első szegmens módosítatlan marad, a második a szegmenshossz értékével előretolódik a Z tengely mentén, a harmadik szegmens a szegmenshossz kétszeresével tolódik előrébb a Z tengely mentén, a negyedik szegmens a szegmenshossz háromszorosával tolódik előre a Z tengely mentén, és így tovább.

Amint ez elkészült, az ObjectBender meghajlítja az objektumot. Ez magyarázható úgy is, hogy a Z tengely képzeletben kör alakúvá válik, s ennek a körnek a középpontja a koordinárendszer sugár,0,0 pontján van, vagyis pozitív érték esetén jobb oldalon, negatív érték esetén bal oldalon. Ha megadtunk blokkhosszt, az elkészült objektum elforgatásra kerül, hogy sínobjektumként használni lehessen.

Az ObjectBendernek utasításokat adhatunk a Coordinates (.B3D) illetve a SetTextureCoordinates (.CSV) parancsok megjegyzéseként. Ezekkel mondhatjuk meg a programnak, hogy a készülő hajlított objektum melyik részére a textúra melyik részét helyezze, avagy mennyivel „léptesse” a Coordinates / SetTextureCoordinates parancsoknál megadott értékeket.

{X=érték}	A megadott értékkel lépteti a szegmenseket a textúrán az X tengely mentén.
{Y=érték}	A megadott értékkel lépteti a szegmenseket a textúrán az Y tengely mentén.

Ezeket a parancsokat a Coordinates / SetTextureCoordinates parancsokkal egy sorba kell írni megjegyzésként, a megjegyzések között bárhova írhatóak. Lásd az alábbi útmutatót további részletekért.

4. Útmutató

Alább egy rövid útmutató olvasható, ami bemutatja, hogy hogyan készítsünk ívelt sínt ObjectBenderrel, hogyan lehet a textúrák koordinátáit beállítani és hogyan lehet hatékonyan dolgozni a parancssoros felülettel.

4.1. Alapok

Tegyük fel, hogy egy 1435 mm nyomtávú, 600 méteres sugarú 25 méteres blokkhosszhoz készülő ívelt sít készítünk. Először is szükség van objektumként egy rövid szegmensre ebből a sínből, mondjuk 5 méternyire.

```
; bal sín
CreateMeshBuilder
AddVertex,-0.7775,0,0
AddVertex,-0.7775,0,5
AddVertex,-0.7175,0,5
AddVertex,-0.7175,0,0
AddFace,0,1,2,3

; jobb sín
CreateMeshBuilder
AddVertex,0.7775,0,0
AddVertex,0.7775,0,5
AddVertex,0.7175,0,5
AddVertex,0.7175,0,0
AddFace,3,2,1,0

; töltés
CreateMeshBuilder
AddVertex,-1,-0.2,0
AddVertex,-1,-0.2,5
AddVertex,1,-0.2,5
AddVertex,1,-0.2,0
AddFace,0,1,2,3
SetColor,150,125,100
```

A fenti kód nem használ textúrákat, ami most némileg megkönnyíti a dolgunkat. Nyissuk meg a program grafikus felületét, válasszuk ki a bemeneti fájlt (itt: szegmens.csv) és állítsuk be a szükséges paramétereket. Mivel a bemeneti objektumunk, vagyis az első szegmens öt méteres és 25 méteres blokkhosszra készül az ív, így öt szegmensre lesz szükségünk, hogy kitöltsük a blokkot. Vagyis:

- Szegmensek száma (Number of segments): 5
- Szegmenshossz (Segment length): 5
- Blokkhossz (Block length): 25
- Sugár (Radius): 600

Ezesetben a nyomtávot nem kell megadni, mert nem használunk túlemelést. Most válasszuk ki a kimeneti fájlt, például jobb_iv_600.csv és kattintsunk a Start gombra. Ha ezután egy *Done!* Feliratú üzenetet kapunk, akkor el is készült az ívelt objektum. Ha akarunk, most készíthetünk egy balos ívet is, ehhez egyszerűen változtassuk a sugarat -600-ra és nyomjuk meg újra a Start gombot.

4.2. Textúrák használata

Tegyük fel, hogy az előbbi sínobjektumban a töltésre textúrát állítunk be:

```
; töltés
CreateMeshBuilder
AddVertex,-1,-0.2,0
AddVertex,-1,-0.2,5
AddVertex,1,-0.2,5
AddVertex,1,-0.2,0
AddFace,0,1,2,3
LoadTexture,ballast.png
SetTextureCoordinates,0,0,0
SetTextureCoordinates,1,0,1
SetTextureCoordinates,2,1,1
SetTextureCoordinates,3,1,0
```

Tudnunk kell, hogy a textúra Y tengelye a valóságban majd a Z tengelynek felel meg. Ha most elképzeljük, hogy az objektum másolatai megjelennek a Z tengely mentén, látjuk, hogy minden egyes másolatot külön-külön fed a textúra, vagyis a textúra minden öt méterben megismétlődik. Ez sokszor jól jöhet, de tegyük fel, hogy mi most olyan textúrával rendelkezünk, amit a teljes, 25 méteres sínre szeretnénk ráhúzni. Ebben az esetben egy szegmensre nem az egész textúra jut, hanem csak annak egy ötöde. Emiatt először a meglévő töltés szegmenset kell módosítanunk:

```
; töltés
CreateMeshBuilder
AddVertex,-1,-0.2,0
AddVertex,-1,-0.2,5
AddVertex,1,-0.2,5
AddVertex,1,-0.2,0
AddFace,0,1,2,3
LoadTexture,ballast.png
SetTextureCoordinates,0,0,0.0
SetTextureCoordinates,1,0,0.2
SetTextureCoordinates,2,1,0.2
SetTextureCoordinates,3,1,0.0
```

A fenti kódban a textúrának csak egy ötöde látszik. Viszont ha most építtetnénk össze az ObjectBenderrel az öt szegmenst, akkor mind az ötön a textúra első egy ötöde látszana! Ez nekünk persze nem felel meg. Nekünk az kell, hogy minden következő szegmensen a textúra következő darabját lássuk a textúra Y tengelye mentén. Ehhez szükséges a bevezetőben említett, a Coordinates vagy jelen esetben SetTextureCoordinates parancs melletti megjegyzésben szereplő utasítás. Adjuk is hozzá a fájlhoz:

```

; töltés
CreateMeshBuilder
AddVertex, -1, -0.2, 0
AddVertex, -1, -0.2, 5
AddVertex, 1, -0.2, 5
AddVertex, 1, -0.2, 0
AddFace, 0, 1, 2, 3
LoadTexture, ballast.png
SetTextureCoordinates, 0, 0, 0.0 ; {Y=0.2}
SetTextureCoordinates, 1, 0, 0.2 ; {Y=0.2}
SetTextureCoordinates, 2, 1, 0.2 ; {Y=0.2}
SetTextureCoordinates, 3, 1, 0.0 ; {Y=0.2}

```

Az ObjectBender most már tudja, hogy minden szegmensnél 0.2-et hozzá kell adnia a SetTextureCoordinates parancsok Y értékeihez. Így az első szegmens természetesen módosíthatatlan marad, de a második már 0.2 és 0.4-es Y értékeket fog használni, a harmadik szegmens 0.4 és 0.6-os, a negyedik szegmens 0.6 és 0.8-as, míg az ötödik, utolsó szegmens 0.8 és 1.0 értékeket fog tartalmazni az Y tengelyre vonatkoztatva. Ezzel a textúra csak minden ötödik szegmens után fog ismétlődni, és mivel 5 méteres szegmessel dolgozunk, ez pontosan kiteszi a megadott 25 méteres blokkhosszt.

4.3. A parancssoros felület használata

Tegyük fel, hogy az ívelt sínhez felsővezetékét készítünk. A felsővezeték nem lehet ívelt, tehát egy egyszerű, 25 méteres vezeték kell létrehozunk. Ha ezt beépítjük az előzőekben elkészített sín-szegmensbe, akkor a sokszorozás és ívelés során a felsővezeték is megsokszorozódna és ívelté válna, ami nyilván nem kívánatos. Ehelyett külön-külön objektumként kell létrehozunk a sánt és a felsővezetékét és majd később építjük őket össze.

Ahhoz, hogy az előbbi ívelt sánt a parancssoros felülettel készítsük el, a következő kódsort kell beírunk a terminálba:

```
ObjectBender szegmens.csv jobb_iv_600.csv /n=5 /s=5 /b=25 /r=600
```

Most készítsük el a felsővezetékét, és mentjük el *felsovezetek.csv* néven:

```

AddVertex, -0.03, 5, 0
AddVertex, -0.03, 5, 25
AddVertex, 0.03, 5, 25
AddVertex, 0.03, 5, 0
AddFace, 3, 2, 1, 0
SetColor, 0, 0, 0

```

A fenti kód egy 25 méter hosszú vezetékét eredményez. Mivel az objektum már eleve 25 méteres, nem kell további másolatokat készítenünk belőle, csak ezt az egy meglevő szegmenst. Ezt hozzá kell adni a *jobb_iv_600.csv* fájlhoz. Ezt megtehetjük a következő módon is:

```
ObjectBender felsovezetek.csv jobb_iv_600.csv /n=1 /s=25 /b=25 /r=600 /a
```

A kódsor végén lévő `/a` kapcsoló arra utasítja az ObjectBendert, hogy ne írja felül a kimeneti fájlt, hanem adja hozzá a művelet eredményét. Most kombináljuk a két parancsot egy batch fájlba. Windows rendszeren ezt elnevezhetjük `start.bat`-nak.

```
ObjectBender szegmens.csv jobb_iv_600.csv /n=5 /s=5 /b=25 /r=600
ObjectBender felsovezetek.csv jobb_iv_600.csv /n=1 /s=25 /b=25 /r=600 /a
```

Ezután ha módosítani akarunk az alap objektumon vagy a felsővezetéken, futtassuk le újra a batch fájlt, és így mindig a legfrissebb állapotot láthatjuk. Hosszabb batch fájlt is létrehozhatunk, hogy többféle, különböző sugarú ívelt sánt készítsünk:

```
; 300m bal
ObjectBender szegmens.csv bal_iv_300.csv /n=5 /s=5 /b=25 /r=-300
ObjectBender felsovezetek.csv bal_iv_300.csv /n=1 /s=25 /b=25 /r=-300 /a
; 300m jobb
ObjectBender szegmens.csv jobb_iv_300.csv /n=5 /s=5 /b=25 /r=300
ObjectBender felsovezetek.csv jobb_iv_300.csv /n=1 /s=25 /b=25 /r=300 /a
; 600m bal
ObjectBender szegmens.csv bal_iv_600.csv /n=5 /s=5 /b=25 /r=-600
ObjectBender felsovezetek.csv bal_iv_600.csv /n=1 /s=25 /b=25 /r=-600 /a
; 600m jobb
ObjectBender szegmens.csv jobb_iv_600.csv /n=5 /s=5 /b=25 /r=600
ObjectBender felsovezetek.csv jobb_iv_600.csv /n=1 /s=25 /b=25 /r=600 /a
```

5. A grafikus felületen megadandó adatok magyarul

Angol	Magyar
Source file	Forrásfájl
Target file	Célfájl
Number of segments	Szegmensek száma
Segment length (m)	Szegmenshossz (méter)
Block length (m)	Blokkhossz (méter)
Radius (m)	Sugár (méter)
Rail gauge (mm)	Nyomtáv (mm)
Initial cant (mm)	Kezdő túlemelés (mm) (az objektum elején)
Final cant (mm)	Végső túlemelés (mm) (az objektum végén)

Remélhetőleg sikerült képet adni a program használatáról.

1 http://openbve.trainsimcentral.co.uk/develop/tool_objectbender.html