

## Erdő generálása a BVEPreproc programmal

Első lépés, hogy elkészítjük a falevél objektumot. Ezeket fogjuk rárakni a faág objektumokra, majd jön a fatörzs...

Ez csak vicc volt.

Elkészítjük/összeollózzuk a textúrákat.

Mivel ez csak egy teszt, három fát rajzoltam.

Felírtam az eredeti képméreteket, magasság/szélesség. Ezekre az adatokra még szükség lesz.

Utána átkonvertáltam 256 színűre, és mindegyiket átméreteztem a textúráknak megfelelő méretekre.

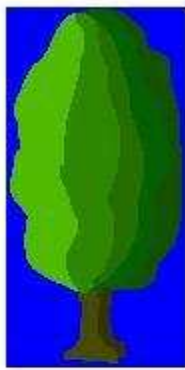
Az egyszerűség kedvéért mindegyiket 64/64 pixelesre.

Egy dologra kell figyelni: A fa a kép közepén helyezkedjen el.

Ha nem így lenne, azt is meg lehetne oldani, de ez nem tárgya ennek a dokumentumnak.



114/63



180/91



183/45



Elkészítjük a \_Fak.h fájlt.

Ebből nem lesz B3D fájl, ezt csak be fogjuk illeszteni a script elejére. Ezért van a .H kiterjesztés.

Az .SRC fájlt lefordítjuk, a .H fájlt csak beillesztjük.

Meg kell adnunk minden egyes fához a következő értékeket.

- A fa átlagos magasságát.
- Az eredeti méretarányt.
- A textúra nevét.
- A transzparens szint.

Ha ezzel készen vagyunk, már nincs sok dolgunk.

```

//-----
//--- Növények típusai. ---
//-----
//--- Mindegyiknek adunk egy számot. Folytonosan 1-től.
cSET( FA_Akac, 1 )
cSET( FA_Koris, 2 )
cSET( FA_Jegenye, 3 )
//--- Beállítjuk a fák darabszámát.
cSET( FA_Db, 3 )

//--- Majd mindegyik fához megadjuk az alábbi adatokat.
//--- A szögletes zárójelek között a fa azonosítója szerepel.
//--- FA_Y[i]: A fa átlagos magassága méterben.
//--- FA_TexYPerX[i]: Az eredeti méretarány.
//--- FA_Tex[i]: A textúra neve.
//--- FA_TransColor[i]: A transzparens szín.

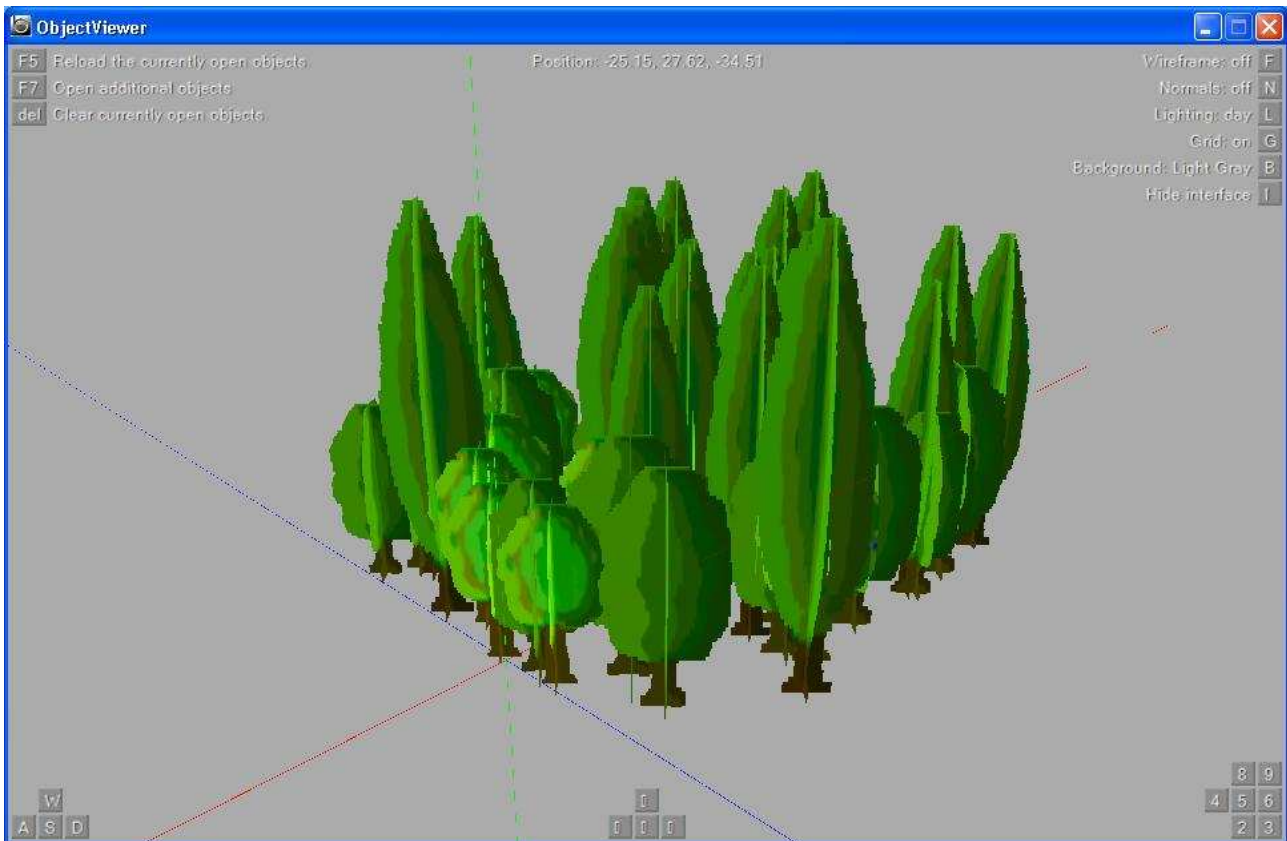
//--- Akác.
cSET( id, &(FA_Akac) )
cSET( FA_Y[&(id)], 10 )
cSET( FA_TexYPerX[&(id)], {114/63} )
cSET( FA_Tex[&(id)], bmp\Fal.bmp )
cSET( FA_TransColor[&(id)], 0,0,255 )

//--- Koris.
cSET( id, &(FA_Koris) )
cSET( FA_Y[&(id)], 12 )
cSET( FA_TexYPerX[&(id)], {180/91} )
cSET( FA_Tex[&(id)], bmp\Fa3.bmp )
cSET( FA_TransColor[&(id)], 0,0,255 )

//--- Jegenye.
cSET( id, &(FA_Jegenye) )
cSET( FA_Y[&(id)], 20 )
cSET( FA_TexYPerX[&(id)], {183/45} )
cSET( FA_Tex[&(id)], bmp\FaJegenye.bmp )
cSET( FA_TransColor[&(id)], 0,0,255 )

```

Ha most legeneráljuk az Erdo.src fájlból a B3D fájlt, ezt kapjuk:



Mit látunk?

Az erdő a Z tengelytől jobbra helyezkedik el.

A középpont a megadott Z érték felénél van. Tükrözéskor ez nagyon jó lesz.

A fák a megadott átlagmagasságtól kis mértékben eltérnek.

Az Erdo.src fájl így néz ki:

```
//=====
//== Erdő. ==
//=====
//--- A fák típusai.
#include( _Fak.h )

//--- A véletlenszám generátor kezdőértéke.
#include( 10 )
//--- Az erdő mérete.
#include( E_X, 30 )
#include( E_Z, 25 )
//--- A fák sűrűsége.
#include( E_Suruseg, 0.1 )

#include( _Erdo.h )
```

Meg kellett adni az erdő szélességét ( E\_X).

Továbbá az erdő hosszát ( E\_Z).

És a fák sűrűségét (E\_Suruseg). Azt jelenti, hogy egy négyzetméteren hány fa legyen.

Ennyi.

Ha most figyelmesen megnézik a képet, és azt, amit ti generáltak, láthatjátok, hogy a kettő

teljesen egyforma.

Néhány szó a véletlenszámok generálásáról:

A számítógép a következő véletlenszámot mindig az előző véletlenszámból képzí.

Viszont az első véletlenszámnak nincs előzője.

A BVEPreproc program induláskor beállítja ezt az értéket - inicializálja a véletlenszám generátort - a mindenkori időnek megfelelően.

Tehát minden futáskor más, ugyanis már eltelt egy kis idő az előzőhöz képest.

De, ott van az

**&INITRND( 10 )**

sor.

Ez azt jelenti, hogy a script beállítja a kezdő értéket 10-re. Innentől kezdve viszont mindig ugyanazt a számsorozatot kapjuk. Tehát az erdő mindig ugyanaz.

Miért is jó ez?

Sajnos a véletlenszerű generálás nem mindig mutat jól. Van amikor nagyon összecsiszognak a fák.

És ha azt szeretnénk, hogy mindig jó legyen, akkor magunknak kell beállítanunk a kezdőértéket.

Ha nem tetszik az eredmény, akkor beállítunk mást, és újra próbálkozunk. Egészen nagy számot is be lehet írni. Akár milliós nagyságrendűt is.

Most vegyék ki (vagy tegyék megjegyzésbe) azt a sort, generáljatok újabb erdőket, és látni fogjátok, hogy mind más.

Próbaképpen kicseréltem a textúrákat, így már sokkal jobban néz ki az erdő.

Még több fajta fával még változatosabb lesz.



Ha valakinek nem tetszik ez a fajta sűrűség megadás, akkor kis módosítással megadható a



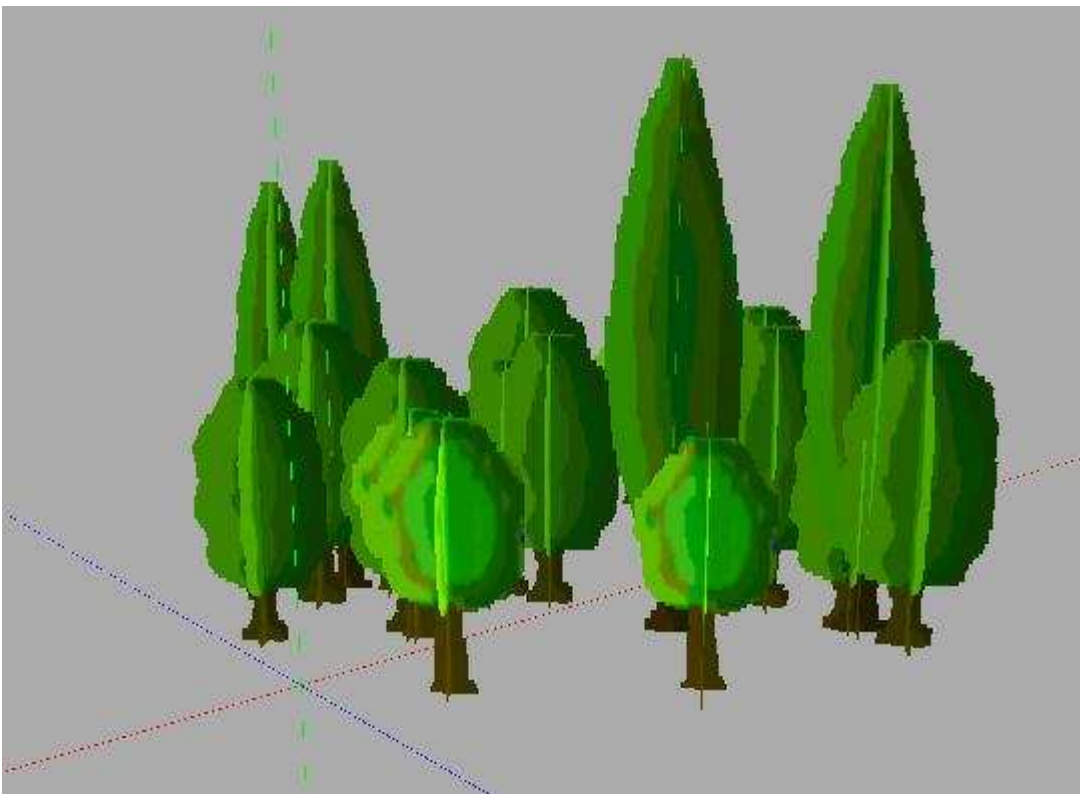
kirakandó fák darabszáma is. Ebből aztán a program számolja ki a sűrűséget. Így működik az Erdo2.src.

```
//=====
//== Erdő 2. ==
//=====
//--- A fák típusai.
#include( _Fak.h )

//--- A véletlenszám generátor kezdőértéke.
//&INITRND( 10 )
//--- Az erdő mérete.
&SET( E_X, 30 )
&SET( E_Z, 25 )
//--- Hány fát szeretnénk kirakni.
&SET( Darabszam, 20 )
//--- A fák sűrűsége.
&SET( E_Suruseg, ( &(Darabszam)/(&(E_X)*&(E_Z)) ) )

#include( _Erdo.h )
```

Az eredmény pedig:



## Második felhasználási mód: Fasor generálás

A fasort, mint azt a neve is mutatja sorban elhelyezkedő fák alkotják. Másik jellegzetessége, hogy többnyire azonos típusú fák alkotják.

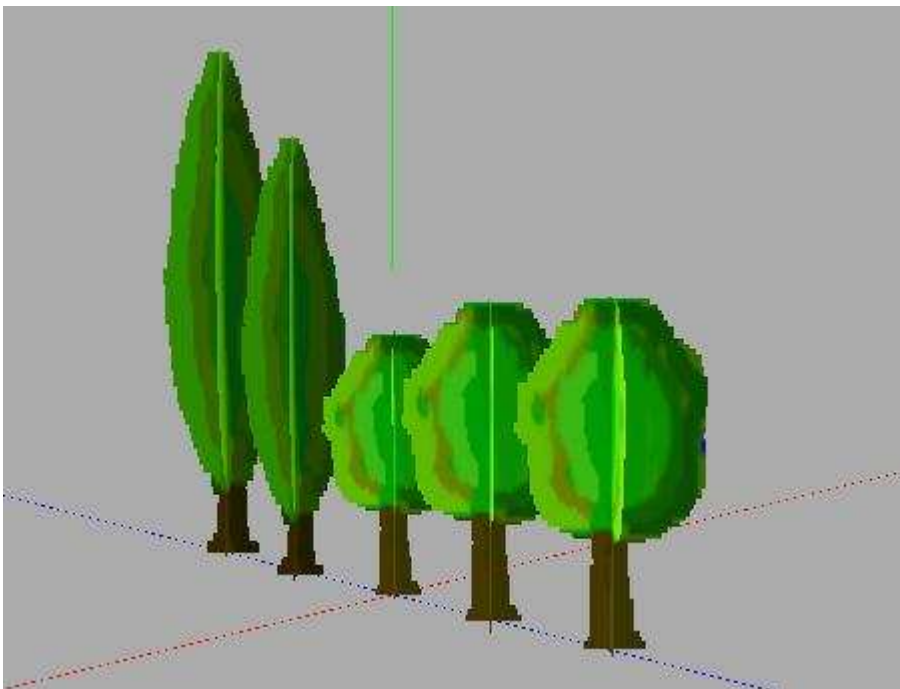
A Fasor.src fájlban csak az elejét másolom be, mert ezzel a résszel szabhatjuk testre a fasorunkat.

```
//=====
//== Fasor. ==
//=====
//--- A fák típusai.
#include( _Fak.h )

//--- A használható fatípusok.
&SET( Arr1, &(FA_Jegenye) )
&SET( Arr2, &(FA_Akac) )
//--- A megadott fa típusok darabszáma.
&SET( ArrDb, 2 )
//--- A fasor hossza.
&SET( E_Z, 25 )
//--- Hány fát szeretnénk kirakni.
&SET( E_Darabszam, 5 )
```

Definiáljuk a használható fa típusokat az Arr1..ArrX változóiban.  
Majd megadjuk ezek darabszámát.  
Továbbá a fasor hosszát.  
És hogy erre a területre hány fát szeretnénk kirakni.  
Lefordítjuk a scriptet.

Az eredmény pedig:



Ha csak egyetlen fajta fát szeretnénk, például csak jegenyét, átírhatjuk a scriptet, de az is megfelelő, ha a második fa helyett is a jegenyét adjuk meg.

```
//--- A használható fatípusok.
&SET( Arr1, &(FA_Jegenye) )
&SET( Arr2, &(FA_Jegenye) )
//--- A megadott fa típusok darabszáma.
&SET( ArrDb, 2 )
```

Súlyozhatjuk is a fák előfordulását így.

```
//--- A használható fatípusok.
&SET( Arr1, &(FA_Jegenye) )
&SET( Arr2, &(FA_Jegenye) )
&SET( Arr3, &(FA_Akac) )
//--- A megadott fa típusok darabszáma.
&SET( ArrDb, 3 )
```

Ekkor kétszer annyi jegenyét várhatunk, mint akácot.

Fontos, hogy legalább két fát meg kell adni, mert különben fordítási hibát kaptok.

A BVEPreproc scripttel a legenerált faszor objektumokat be is helyezhetjük a pályánkba.

Például van két faszor objektumunk, a számuk 100 és 101.

Ezeket szeretnénk változtatni de úgy, hogy lyukak is lehessenek köztük.

Ezt a kódsort beilleszthetjük a route fájlunkba.

```
//--- A használható faszor objektumok.
&SET( Arr1, 100 )
&SET( Arr2, 101 )
&SET( Arr3, 0 )
//--- A megadott objektumok darabszáma.
&SET( ArrDb, 3 )

//--- Kezdő, és záró szelvényszám.
&SET( nStart, 1012.5 )
&SET( nEnd, 1500 )

//--- Kirakó ciklus.
&SET( nCtr, &(nStart) )
&WHILE( ( &(nCtr) < &(nEnd) ) )
  //--- Mit kell kirakni?
  &SET( nId, &( Arr&RND( &(ArrDb) ) ) )
  //--- Ha nagyobb mint nulla.
  &IF( ( &(nId) > 0 ) )
    //Pozíció.
    &(nCtr),
    //Az objektum.
    .FreeObj 0; &(nId); -5; 0; 0,
  &END()
  //--- A szelvényszám növelése.
  &SET( nCtr, ( &(nCtr)+25 ) )
&END()
```

A nullával definiált objektum a lyuk.

Természetesen itt is súlyozhatunk.

Na de nézzük az eredményt:

```
|1012.5,  
.FreeObj 0; 100; -5; 0; 0,  
1037.5,  
.FreeObj 0; 101; -5; 0; 0,  
1112.5,  
.FreeObj 0; 101; -5; 0; 0,  
1137.5,  
.FreeObj 0; 100; -5; 0; 0,  
1162.5,  
.FreeObj 0; 100; -5; 0; 0,  
1187.5,  
.FreeObj 0; 101; -5; 0; 0,  
1212.5,  
.FreeObj 0; 101; -5; 0; 0,  
1237.5,  
.FreeObj 0; 101; -5; 0; 0,  
1287.5,  
.FreeObj 0; 101; -5; 0; 0,  
1387.5,  
.FreeObj 0; 101; -5; 0; 0,  
1412.5,  
.FreeObj 0; 101; -5; 0; 0,  
1437.5,  
.FreeObj 0; 100; -5; 0; 0,  
1462.5,  
.FreeObj 0; 101; -5; 0; 0,
```

## Harmadik felhasználási mód.

A fa generátort másként is felhasználhatjuk.

Tegyük fel, hogy az állomásépület objektumunk mellé szeretnénk egy jegenyefát rakni. Ennyit kell tenni.

Az objektum elejére beilleszteni a következő sort.

```
&INCLUDE( _Fak.h )
```

Majd beállítani a változókat, és meghívni a generátort.

```
&SET( Fa_Type, &(FA_Jegenye) )  
&SET( FaCommand, Translate 10, 0, 5 )  
&INCLUDE( _FaRandom.h )
```

Ha azt sem akarjuk megmondani, hogy milyen fát szeretnénk. Akkor ez is elég.

```
&SET( FaCommand, Translate 10, 0, 5 )  
&INCLUDE( _FaRandom.h )
```

Ilyenkor véletlenszerű a típusa is.

Ne felejtjük el az objektum fájlt lefordítani!

## Haladó felhasználóknak:

Az \_FaRandom.h fájlban van egy ilyen sor.

```
&SET( %fdelta, 0.15 )
```



Ez mondja meg, hogy a generált fák magassága hányad résszel térhet el a megadott átlagmagasságtól pozitív és negatív irányba.

Az akác esetében ez  $10 * 0.15 = 1.5$

Tehát a generált akác fák magassága 8.5 és 11.5 méter között ingadozhat.

Megjegyzés:

Az erdő generálásnak ez a legegyszerűbb módja, de ki lehet találni jobb módszert is. Az \_Erdo.h fájl kell máshogy megírni.

Használjátok megelégedéssel!

Tomasz