

## openBVE pályakészítés Leírás a pályakészítéshez használható parancsokról

A leírás az **openBVE**-hez készíthető pályák **CSV kiterjesztésű route fájljaiban** használható parancsok magyarázatát írja le.

A dokumentum forrása a [http://openbve.trainsimcentral.co.uk/develop/route\\_csv.html](http://openbve.trainsimcentral.co.uk/develop/route_csv.html) weboldal. Egyes elemek működése ismeretlen, vagy még nem kipróbált, vagy a leírás még hiányzik. Ezek **sárga színű kiemeléssel** vannak jelezve. Ezért a leírás nem teljes körű, néhol hiányos, vagy még angol nyelvű. A **zöld kiemelésű leírások** eredeti leírása még nem ismert, az így jelölt leírások csak kipróbálások, illetve a korábbi BVE programokban alkalmazott lehetőségek eredményei.

Az **openBVE** folyamatos frissítése miatt előfordulhatnak olyan parancsok, amelyek nem a leírt módon működnek.

Utolsó frissítés 2009.06.02.

### Tartalomjegyzék

<b>Általános tudnivalók, fogalmak .....</b>	<b>4</b>
route fájl .....	4
blokk .....	4
alapértelmezett érték.....	4
0. vágány (nullás vágány).....	4
egyenes vonalú vágány.....	4
origó .....	4
szelvényezés, szelvényszám.....	4
nyomtáv.....	4
sínkoronaszint .....	5
vágánytengely .....	5
vágánytengelytávolság.....	5
<b>Parancsfelépítés .....</b>	<b>5</b>
Megjegyzés .....	5
Parancsok.....	5
A „With” meghatározás .....	7
Szelvényszám .....	7
<b>Helyettesítő kifejezések .....</b>	<b>8</b>
\$Chr .....	8
\$Rnd .....	8
\$Sub .....	9
Folyamat példa.....	9
<b>Parancsok.....</b>	<b>10</b>
„Options” szakasz .....	10
Options.UnitOfLength .....	10
Options.UnitOfSpeed .....	10
Options.BlockLength .....	11
Options.ObjectVisibility .....	11
Options.SectionBehavior.....	11
Options.FogBehavior .....	11
„Route” szakasz .....	12
Route.Comment .....	12
Route.Image .....	12
Route.Timetable.....	12
Route.Change.....	12
Route.Gauge .....	12
Route.Signal .....	12
Route.RunInterval .....	13
Route.AccelerationDueToGravity .....	13
Route.Elevation .....	13
Route.Temperature.....	13
Route.Pressure .....	13
Route.AmbientLight .....	14

Route.Directionallight .....	14
Route.LightDirection .....	14
Route.DeveloperID.....	14
„Train” szakasz .....	15
Train.Folder; Train.File .....	15
Train.Run; Train.Rail .....	15
Train.Flange.....	15
Train.Timetable (Day).....	15
Train.Timetable (Night).....	15
Train.Gauge .....	16
Train.Interval .....	16
„Structure” szakasz.....	17
Structure.Pole .....	18
„Texture” szakasz .....	18
Texture.Background .....	18
„Cycle” szakasz.....	19
Cycle.Ground.....	19
„Signal” szakasz.....	20
„Track” szakasz.....	21
Vágányok: .....	21
Track.Railstart .....	21
Track.Rail.....	21
Track.Railtype .....	21
Track.Railend .....	22
Track.Accuracy .....	22
Track.Adhesion .....	22
Vonalvezetés: .....	22
Track.Curve .....	22
Track.Pitch .....	22
Track.Turn.....	23
Track.Height.....	23
Objektumok:.....	23
Track.Freeobj .....	23
Track.Wall .....	24
Track.Wallend .....	24
Track.Dike .....	24
Track.Dikeend .....	24
Track.Pole .....	25
Track.Poleend .....	25
Track.Crack.....	25
Track.Ground .....	25
Állomások:.....	25
Track.Sta .....	25
Track.Station .....	26
Track.Stop .....	27
Track.Form .....	27
Jelzők:.....	28
Track.Limit .....	28
Track.Section.....	28
Track.SigF .....	29
Track.Signal .....	29
Track.Relay.....	30
Biztonsági rendszerek: .....	30
Track.Beacon.....	30
Track.Transponder .....	30
Track.AtsSn .....	30
Track.AtsP .....	30
Track.Pattern .....	30
Track.PLimit .....	30
Egyéb:.....	30
Track.Back.....	30
Track.Fog.....	31

Track.Brightness.....	32
Track.Marker.....	32
Track.PointOfInterest.....	32
Track.PreTrain.....	32
Track.Announce.....	32
Track.Doppler.....	32
Track.Buffer.....	32
<b>Összefoglaló.....</b>	<b>33</b>
Szelvénytípus.....	33
Helyettesítők, véletlenszám generálók.....	33
„Options” szakasz.....	33
„Route” szakasz.....	33
„Train” szakasz.....	33
„Structure” szakasz.....	33
„Texture” szakasz.....	34
„Cycle” szakasz.....	34
„Signal” szakasz.....	34
„Track” szakasz.....	34
<b>Gyakoribb hibáüzenetek.....</b>	<b>36</b>
Route fájl hibák.....	36
<i>Típus Fájlnév</i> not found.....	36
Index is required to reference a non-existing rail in Track.Railstart.....	36
IdxStType references an object not loaded in Track. <i>parancsnév</i> .....	36
The number of aspects is not supported in Track.Signal.....	36
A stop without a station is invalid in Track.Stop.....	36
Objektum hibák.....	36
The Command <i>parancs</i> is not supported; The attribute <i>parancs</i> is not supported.....	36
The direction indicated by X, Y and Z is expected to be non-zero in <i>parancs</i> .....	36
At most <i>x</i> arguments are expected in <i>parancs</i> .....	36
0 arguments are expected in [Meshbuilder].....	36
<b>Egyéb fogalmak.....</b>	<b>37</b>
űrszelvény.....	37
kitérő.....	37
túlemelés.....	37
<b>Példák.....</b>	<b>38</b>
Példa az Options.ObjectVisibility módjainak különbségeiről.....	38
Példa a Track.Brightness parancs alkalmazására.....	38
Példa a Track.Stop parancs alkalmazására.....	38
Példa a Route.Signal, Track.SigF és Track.Section használatára.....	38
1. „Route” szakaszba tartozó parancsok beírása.....	38
2. Jelzési képek textúráinak elkészítése.....	38
3. Jelzési képek objektumainak elkészítése.....	39
4. A jelzéstípusok definiálása a Structure szakaszban.....	39
5. A Track.SigF és Track.Sextion parancsok alkalmazása a „Track” szakaszban.....	39

## Általános tudnivalók, fogalmak

### route fájl

csv kiterjesztésű fájl, amely a játszható útvonalat tartalmazza. Ezt a fájlt kell elindítani az openBVE-ben.

A csv kiterjesztésű **route fájlban** szövegalapú kifejezéseket kell adni a pálya megépítéséhez.

A route fájlban olyan mappában kell lennie, amelynek a szülőmappája tartalmazza a **Railway** és **Train** mappákat. A fájl neve tetszőleges lehet, de a kiterjesztése kötelezően **csv**. A fájl kiértékelése soronként történik fentről lefelé, az egyes sorokban több kifejezést is lehet írni, amelyek kiértékelése balról jobbra történik.

A route fájlban használatos objektumokat parancsokkal kell definiálni („Structure” szakasz). Ezen felül a pályához definiálható egy alapértelmezett jármű, valamint kép is. Ezt követően a pálya felépítését kell megadni különböző parancsokkal („Track” szakasz). Ebben a szakaszban szelvényszámokat kell megadni, amelyben megadható többek között, hogy merre és milyen irányú ív legyen, hol legyenek az állomások, stb. Általában a „Track” szakasz parancsait csak azután kell használni, miután a többi szakasz parancsait használtuk (definiáltuk az objektumokat).

Az openBVE-ben egy **route fájlban** csak egy útvonal készíthető. Ha az adott útvonalat visszafelé is el akarjuk készíteni, akkor csak egy másik route fájlban van erre lehetőség, amelyben újra fel kell építeni a pályát az ellenkező iránynak megfelelően.

### blokk

Az openBVE-ben a pálya **blokkokból** épül fel. A blokkok [vágánytengelyének](#) kezdő és végpontját az openBVE **egyenes vonallal köti össze**. Emiatt ívekben az ív külső oldalán elhelyezett objektumon hézagok lehetnek. Néhány parancs csak blokk kezdetén alkalmazható, és a blokk elején beállított tulajdonsága csak a következő blokk elején változtatható. Nagyobb méretű blokk használata esetén előfordulhat, hogy íves objektumokat kell készíteni. Kisebb méretű blokkok használatával kevesebb íves objektumot kell készíteni.

### alapértelmezett érték

Amennyiben egy általános parancsot nem adunk meg a route fájlban vagy egy parancsnak bármely paraméterét nem adjuk meg, akkor mindig az **alapértelmezett értéket** veszi figyelembe a program. Az alapértelmezett értékek az egyes parancsoknál fel vannak tüntetve.

### 0. vágány (nullás vágány)

Az egy útvonalnak megfelelően a járművünk mindig csak egy útirányon haladhat. A járművünk mindig a **„nullás vágányon”** (továbbiakban: „0. vágány”) halad

### egyenes vonalú vágány

A blokkok [vágánytengelyének](#) kezdő és végpontját az openBVE **egyenes vonallal köti össze**. Az íves kialakítású sínobjektumokban a sínszálak, illetve a vágánytengely helyzete eltér az egyenes vonaltól. Az ívekben az objektumok távolságát mindig az egyenes vonalnak megfelelő vágánytengely helyzetéhez képest kell meghatározni.

### origó

A koordinátatengelyek metszéspontját **origónak** nevezzük. A BVE-ben az objektumokat gyakorlatilag egy koordináta-rendszer szerint kell elhelyezni. A BVE-ben az origó a **0. vágány vágánytengelyének** (X tengely), a **sín koronaszintjének** (Y tengely), és az **aktuális szelvényszám** (Z tengely) a metszéspontja. Az objektumok elhelyezését ennek megfelelően kell meghatározni.

### szelvényezés, szelvényszám

A vasúti pálya helyének meghatározása a kezdőponttól mért távolságokkal történik, ez a **szelvényezés**.

Az openBVE-ben **alapértelmezésben méterben** kell megadni a szelvényszámokat.

### nyomtáv

**Nyomtáv (nyomtávolság)** vagy **nyomköz** a vágány két sínszálának egymástól való távolsága a sínfejek belső élei között a vágánytengelyre merőlegesen mérve.

**Normál nyomtávolság:** Angliában 4 láb 8,5 hüvelykben állapították meg, amely **1435 mm**. A világ legtöbb vasútja normál nyomtávolságú.

Eltérő példák:

**Széles nyomtávolság:** 1520 mm Oroszországban

**Keskeny nyomtávolság:** 760 mm a magyarországi legtöbb kisvasút nyomtávolsága

## sínkoronaszint

A sín felső érintősíkjának abszolút magassága (íves pályán a belső sínszálon mérve).

## vágánytengely

A [nyomtávolság](#) felezőpontjainak vonala. (Ívben nyombővítés esetén a külső sínszáltól mért fél nyomtávolságra fekszik.)

## vágánytengelytávolság

A szomszédos [vágánytengelyek](#) távolsága.

## Parancsfelépítés

A route fájl soraiba egyéni **kifejezéseket** (parancsokat) lehet írni, amelyeket vesszővel kell elválasztani.

```
Kifejezés1, Kifejezés2, Kifejezés3, ..., Kifejezésn
```

A **kifejezéseket** a következő formákban lehet megadni:

### Megjegyzés

```
;Megjegyzés
```

Egy megjegyzés/leírás írható be, ha a sor elejére pontosvesszőt írunk. A pontosvesszővel kezdődő sorokat az openBVE nem értékeli ki, azaz figyelmen kívül hagyja.

```
;Ez egy megjegyzés.
```

### Parancsok

Parancs, **paraméter nélkül:**

```
Parancsnév
```

Parancs, **paraméterekkel:**

```
Parancsnév paraméter1;paraméter2;paraméter3;...;paramétern
```

```
Parancsnév(paraméter1;paraméter2;paraméter3;...;paramétern)
```

Parancs, **indexszámokkal és utótagokkal:**

```
Parancsnév(Index1;Index2;...;Indexn) Paraméter1;Paraméter2;...;Paramétern
```

```
Parancsnév(Index1;Index2;...;Indexn).Utótag Paraméter1;Paraméter2;...;Paramétern
```

```
Parancsnév(Index1;Index2;...;Indexn).Utótag(Paraméter1;Paraméter2;...;Paramétern)
```

### Szabályok:

A parancs a parancsnévvel kezdődik, és a paraméterekkel folytatódik. Az indexszámokat és a paramétereket **pontosvesszővel** kell elválasztani. Amennyiben indexszámot kell használni, akkor az **indexszámot** zárójelbe kell tenni.

A leírt szintaktikai variációkból bármelyik alkalmazható. Azonban a [\\$Chr](#), [\\$Rnd](#) és [\\$Sub](#) helyettesítő kifejezéseket nem lehet használni, mert ezeknél a kifejezéseknél a paramétereket zárójelbe kell tenni. Ezekről a kifejezésekről az [„Helyettesítő kifejezések”](#) témakörben találhatóak információk.

Két szintaktikai variáció alkalmazható: az egyikben a **Parancsnév** és az első **Paraméter** közé **szóközt** kell tenni. A másikban a **Parancsnév** után **zárójelbe kell tenni a Paramétereket**, illetve az **Utótag** használata kötelező.

A továbbiakban:

azok a paraméterek **zöld** színnel vannak jelölve, amelyek mértékegysége az [Options.UnitOfLength](#) beállításától függ.

azok a paraméterek **kék** színnel vannak jelölve, amelyek mértékegysége az [Options.UnitOfSpeed](#) beállításától függ.

ha egy **Utótag** kötelező akkor az **félkövérrel**, ha elhagyható, akkor azok **szürkével** van jelölve.

*Példa: (a következő parancsok ugyanazt jelentik):*

```
Track.Stop 1; 10; 10
```

```
Track.Stop(1; 10; 10)
```

*Példa, utótag nélkül:*

```
Structure.Rail(0) betonalj.b3d
```

*Példa, utótaggal:*

```
Structure.Rail(0).Load betonalj.b3d  
Structure.Rail(0).Load(betonaj.b3d)
```



## A „With” meghatározás

### With szakasz

**Szakasz:** annak a szakasznak a neve, amelyhez a parancsokat kívánjuk megadni.

Az ezt követő parancsok a megadott szakaszhoz tartoznak.

*Példa a „Route” szakasz parancsaihoz:*

```
With Route
.Gauge 1435
.RunInterval 600
.Temperature 24
.Pressure 100.935
```

Ez megegyezik a következő bejegyzésekkel:

(„With” meghatározás nincs, de ekkor minden parancs elé a **szakasz** nevét be kell írni):

```
Route.Gauge 1435
Route.RunInterval 600
Route.Temperature 24
Route.Pressure 100.935
```

## Szelvénytáv

### Szelvénytáv

Egy **szám**, amely a [szelvénytávot](#) határozza meg. Az ezt követő parancsok az adott szelvénytávba kerülnek elhelyezésre. A [„Track” szakasz](#) parancsainak alkalmazása előtt általában egy szelvénytávot kell meghatározni. Az **alapértelmezett mértékegysége méter**, de ez az [Options.UnitOfLength](#) parancssal változtatható.

**Rész<sub>1</sub>:Rész<sub>2</sub>:...:Rész<sub>n</sub>**

Egy bonyolultabb változata a szelvénytáv megadásának, amelyben az [Options.UnitOfLength](#) parancssal van kapcsolatban, a következők szerint: a **Rész<sub>1</sub>** a **Faktor<sub>1</sub>** többszörözését, a **Rész<sub>2</sub>** a **Faktor<sub>2</sub>** többszörözését jelenti, stb. A részeket **kettősponttal** kell elválasztani.

Például, ha **mérföld**, **lác**, **láb** mértékegységeket szeretnénk használni, akkor az [Options.UnitOfLength](#) parancshoz írjuk be a következőt:

```
Options.UnitOfLength 1609.344; 20.1168; 0.3048
```

Így az egyes paraméterekbe írt szám akkora távolságot fog jelenteni, amennyit beállítottunk.

*Példa:*

**1:2:3** = 1 mérföld, 2 lác, 3 láb =  $1 \times 1609.344 + 2 \times 20.1168 + 3 \times 0.3048 = 1650.492$  méter.

Ha a távolságba (szelvénybe) nem írjuk be az összes paramétert, akkor mindig jobbról kezdve értendőek az értékek, és a maradék paraméterek helyét 0-nak kell tekinteni.

*Példa:*

**0:0:5** = Ez rövidhető 0:5 beírásával, vagy csak az utolsó paraméter (5) beírásával, a végeredmény mindegyik esetben 5 láb =  $5 \times 0.3048$  méter = 1.524 méter.

Lásd még az [Options.UnitOfLength](#) parancs leírását.

## Helyettesítő kifejezések

Mielőtt a route fájlban levő parancsokat a program ténylegesen értelmezi, a beírt kifejezéseket először feldolgozza. Először a **\$Chr**, **\$Rnd** és **\$Sub** helyettesítésekkel beírtakat cseréli a megfelelő kifejezésekre. A szintaxisokat nem lehet szabadon választani, azokat az alábbi formákban kell beírni:

### \$Chr

#### \$Chr(Ascii)

**Ascii:** Egy egész szám, 20 és 127 között, valamint 10 vagy 13 lehet, amely az **ASCII** karakterkészlet számát mutatja. További információ az **ASCII** karakterkészletről: <http://hu.wikipedia.org/wiki/ASCII>

Ez a helyettesítő kicseréli a számot a számhoz tartozó **ASCII** karakterre.

Példák:

ASCII-kód	Jelentés	Karakter
10	Új sor	<i>új sor</i>
13	Új sor	<i>új sor</i>
40	Kezdő zárójel	(
41	Végződő zárójel	)
44	Vessző	,
59	Pontosvessző	;

Ezeket akkor célszerű használni, ha olyan karakterre van szükségünk, amelyeket az openBVE a szintaxisban használ, és hibák adódhatnak a használatukból.

#### *Példa a \$Chr helyettesítő használatához:*

Route.Comment Ez az én pályám.\$Chr(13)Ez a mondat egy új sorban kezdődik\$Chr(44)és ez a mondatrész egy vessző után következik. \$Chr(40)Ez a mondat pedig zárójelben van.\$Chr(41)

#### *Ez a szöveg a következőképpen jelenik meg az openBVE menüjében, a pálya leírásaként:*

Ez az én pályám.

Ez a mondat egy új sorban kezdődik, és ez a mondatrész egy vessző után következik.  
(Ez a mondat pedig zárójelben van.)

### \$Rnd

#### \$Rnd(Minimum; Maximum)

Ez a helyettesítő egy véletlenszerű számot tesz be, amely nem kisebb a **Minimum** értéknél és nem nagyobb a **Maximum** értéknél.

**Minimum:** Értéke egy szám, amely a számunkra megfelelő tartomány **legkisebb** értéke.

**Maximum:** Értéke egy szám, amely a számunkra megfelelő tartomány **legnagyobb** értéke.

Ez a helyettesítő egy véletlenszámot generál a megadott tartományból. Ezeket a route fájlban a parancsokba lehet írni, ha például olyan objektumot szeretnénk elhelyezni, ami véletlenszerű legyen, vagy például véletlenszerű helyen jelenjen meg.

#### *Példa a \$Rnd helyettesítő használatához:*

10\$Rnd(3;5)0, Track.FreeObj 0; 1

*A példából származó lehetséges variációk: (A szelvényszám harmadik számjegye **3, 4, vagy 5** lehet).*

1030, Track.FreeObj 0; 1

1040, Track.FreeObj 0; 1

1050, Track.FreeObj 0; 1



**\$Sub****\$Sub(Index) = Érték****Index:** Egy szám, amely a változó indexszámát jelenti.**Érték:** Egy definiált változó.

A használandó **Értéket** az **Indexként** definiáltként alkalmazza. Ez olyankor hasznos, ha egy véletlenszám tartományt többször szeretnénk használni.

**\$Sub(Index)****Index:** Egy szám, amely a változó indexszámát jelenti.Ez a helyettesítő felváltja azt, ami az **Indexben** definiálva van.*Példa az említett két helyettesítő alkalmazására:*

```
$Sub(0) = $Rnd(3;5)
1000, Track.FreeObj $Sub(0); 47
1020, Track.FreeObj $Sub(0); 47
1040, Track.FreeObj $Sub(0); 47
```

*Ebben a példában mindhárom **Track.Freeobj** parancs ugyanazt a véletlenszám helyettesítést használja, így mindhárom objektum **biztosan ugyanazon a vágányon** fog megjelenni.*

*Amennyiben egyénileg kicseréljük mindhárom sorban a **\$Sub(0)** helyettesítőt a meghatározott **\$Rnd(3;5)** helyettesítőre, akkor az is lehetséges, hogy az objektumok **más-más vágányon** fognak megjelenni.*

**\$Sub(Index; Érték)****Index:** Egy szám, amely a változó indexszámát jelenti.**Érték:** Egy definiált változó, az adott **Indexhez** ezentúl az **Értéket** rendeli hozzá a program.

Akkor alkalmazzuk, ha az **Indexszámot** változatlanul szeretnénk hagyni, de a hozzárendelt **Értéket** meg akarjuk változtatni.

**Folyamat példa**

A **\$Chr**, **\$Rnd** és a **\$Sub** helyettesítők feldolgozása után a [route fájlban](#) lévő kifejezések a [szelvénytípus](#) szerint növekvő sorrendbe rendeződnek.

*Példa részlet egy route fájlból:*

```
1000, Track.FreeObj 0; 23
1050, Track.RailType 0; 1
10$Rnd(3;7)0, Track.Rail 1; 4
Track.FreeObj 1; 42
```

*A feldolgozás folyamán a program a **\$Rnd** helyettesítőt véletlenszerűen a „3”-as számra cseréli:*

```
1000, Track.FreeObj 0; 23
1050, Track.RailType 0; 1
1030, Track.Rail 1; 4
Track.FreeObj 1; 42
```

*Ezt követően a parancsok [szelvénytípus](#) szerint növekvő sorrendbe rendeződnek:*

```
1000, Track.FreeObj 0; 23
1030, Track.Rail 1; 4
Track.FreeObj 1; 42
1050, Track.RailType 0; 1
```

## Parancsok

### „Options” szakasz

Ebben a szakaszban az egész route fájlra vonatkozó általános beállításokat adhatunk meg. Ezek a parancsok a későbbi parancsokra hatnak, ezért ezt a szakaszt kell először a route fájlba írni.

#### Options.UnitOfLength

**Options.UnitOfLength** *FaktorMéter<sub>1</sub>; FaktorMéter<sub>2</sub>; ...; FaktorMéter<sub>n</sub>*

Ezzel a paranccsal az **út** (távolság) mértékegységét lehet részletezve beállítani.

**FaktorMéter**: Egy **szám** (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely meghatározza, hogy hány **méter** hosszú az adott **faktor**, amely a [szelvényezéshez](#) használt távolságot fogja jelenteni. Alapértelmezésben az első paraméter **1**, a többi paraméter **0**.

Ez a parancs a többi parancsra is hatással van!

Amennyiben csak 1 mértékegységet szeretnénk használni, akkor csak **egy paramétert** adjunk meg. Például ha méter helyett **yardban** (1 yard = 0,9144 méter) akarjuk a pályát elkészíteni, akkor a következőt kell beírni:

```
Options.UnitOfLength 0.9144
```

További példák:

A távolság egysége	FaktorMéter
mérföld	1.609344
lác	20.1168
méter	1
yard	0.9144
láb	0.3048

Amennyiben több mértékegységet szeretnénk használni, amelyek például egymás váltószámai, akkor több **paramétert** is megadhatunk.

A továbbiakban azok a paraméterek **zöld** színnel vannak jelölve, amelyek mértékegysége az **Options.UnitOfLength** beállításától függ.

Ezt a parancsot a többi parancs előtt dolgozza fel a program. Egy [route fájlban](#) csak egyszer adható meg.

#### Options.UnitOfSpeed

**Options.UnitOfSpeed** *FaktorKMH*

**FaktorKMH**: Egy **szám** (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely meghatározza, hogy hány **kilométer/óra** legyen a sebesség egysége. Az alapértelmezett érték **1**.

Példák:

A sebesség egysége	FaktorKMH
méter/szekundum	3.6
mérföld/óra	1.609344
kilométer/óra	1

A továbbiakban azok a paraméterek **kék** színnel vannak jelölve, amelyek mértékegysége az **Options.UnitOfSpeed** beállításától függ.

Ezt a parancsot a többi parancs előtt dolgozza fel a program. Egy [route fájlban](#) csak egyszer adható meg.

---

## Options.BlockLength

### Options.BlockLength *Hossz*

**Hossz:** Egy **szám** (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely egy **blokk** hosszának értékét állítja be. Alapértelmezésben egy blokk **25 méter** hosszú.

Ez a parancs a **blokk** hosszát állítja be, az **egész route fájlra** vonatkozóan, ezért egy route-fájlban csak egyszer kell megadni.

Ez a parancs a többi parancsra is hatással van!

Ezt a parancsot a többi parancs előtt dolgozza fel a program. Egy [route fájlban](#) csak egyszer adható meg.

---

## Options.ObjectVisibility

### Options.ObjectVisibility *Mód*

**Mód:** Az objektumok láthatóságának módja. Az alapértelmezett érték **0**.

**0:** Az objektumok láthatatlanok lesznek, amint teljesen áthalad a kamera azon a blokkon, ahol az objektum elhelyezésre került.

**1:** Az objektumok csak akkor tűnnek el, ha a kamera már teljes terjedelmében áthaladt az objektumon.

A játék során a jármű haladásával együtt új objektumok tűnnek fel a látómezőben, és azok az objektumok láthatatlanná válnak, amelyek mellett a járművünk már elhaladt.

A „**0**” beállítással sok olyan objektum, vagy annak bizonyos részei láthatatlanná válhat (menet közben eltűnhet), amelyek mellett a járművünk még nem haladt el. Egyes objektumokat így csak korlátozottan lehet alkalmazni, vagy pedig más formában is el kell készíteni.

Az „**1**” beállítással azonban kizárólag csak akkor tűnnek el az objektumok, amikor a járművünkkel elhaladtunk mellette, így az objektumokat bárhogyan elkészíthetjük.

Ezt a parancsot a többi parancs előtt dolgozza fel a program. Egy [route fájlban](#) csak egyszer adható meg.

[Példa az Options.ObjectVisibility módjainak különbségeiről](#)

---

## Options.SectionBehavior

### Options. SectionBehavior *Mód*

**Mód:** A térközműködést lehet beállítani a pályára vonatkozóan. Az alapértelmezett érték **0**.

**0:** Index-alapú térközműködés.

**1:** Érték-alapú térközműködés.

A parancs a **Track.Section** parancsokkal létrehozott térközök működését állítja be.

[Részletes leírását lásd a Track.Section parancs leírásánál.](#)

[Példa a Route.Signal, Track.SigF és Track.Section használatára](#)

Ezt a parancsot a többi parancs előtt dolgozza fel a program. Egy [route fájlban](#) csak egyszer adható meg.

---

## Options.FogBehavior

### Options. FogBehavior *Mód*

**Mód:** A [Track.Fog](#) parancsok működési módját állítja be. Az alapértelmezett érték **0**.

**0:** **Blokk-alapú működési mód.** A szín és a távolság lineárisan változik az első olyan [blokk](#) kezdetétől a blokk végig, ahol a [Track.Fog](#) parancs újra alkalmazva van.

**1:** **Lineáris működési mód.** A szín és a távolság lineárisan változik két szomszédos [Track.Fog](#) parancs beállításainak függvényében.

## „Route” szakasz

Ebben a szakaszban az egész route fájlra vonatkozó általános leírásokat, információkat adhatunk meg.

### Route.Comment

#### Route.Comment Szöveg

**Szöveg:** Az ide beírt szöveg jelenik meg a route fájl leírásaként.

A **Szövegben** nem használható vessző, és új sort sem lehet írni. Erre a célra a **\$Chr** kifejezést kell használni.

A **\$Chr** helyettesítőt kell használni, ha új sort, vesszőt, pontosvessző írásjeleket szeretnénk tenni a szövegbe.

### Route.Image

#### Route.Image Fájlnev

**Fájlnev:** Az itt megadott képfájl az openBVE menüben a pálya képeként jelenik meg. Relatív útvonalon kell megadni a route fájlhoz képest.

### Route.Timetable

#### Route.Timetable Szöveg

**Szöveg:** Ez a szöveg fog megjelenni a **menetrend** címsoraként.

Ha a **Train.TimeTable** parancs nincs alkalmazva, akkor egy alapértelmezett menetrend jeleníthető meg, amely az állomások neveit, érkezési és indulási időpontokat tartalmazza. A **Szöveg** ezen menetrend tetején jelenik meg.

A **\$Chr** helyettesítőt kell használni, ha új sort, vesszőt, pontosvessző írásjeleket szeretnénk tenni a szövegbe.

### Route.Change

#### Route.Change Mód

A pálya indításakor a biztonsági berendezés állapotát állítja be.

**Mód:** A biztonsági berendezés állapota.

**-1:** A biztonsági berendezés **aktiválva van** és a jármű **üzemi fékkel** van fékezve.

**0:** A biztonsági berendezés **nincs aktiválva** és a jármű **vészfékkel** van fékezve.

**1:** A biztonsági berendezés **aktiválva van** és a jármű **vészfékkel** van fékezve.

### Route.Gauge

#### Route.Gauge ÉrtékMilliméter

**ÉrtékMilliméter:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a route fájlban használt **nyomtávot** jelenti **milliméterben**. Az alapértelmezett érték **1435**.

A **Route.Gauge** parancs megegyezik a **Train.Gauge** parancssal.

### Route.Signal

#### Route.Signal(Index).Set Sebesség

A .Set utótag nem kötelező.

**Index:** Egy szám, amely adott jelzéképet jelent, és ehhez sebességértéket lehet hozzárendelni.

**Sebesség:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a jelzéképhez tartozó maximális engedélyezett sebességet jelenti, **km/h**-ban. Alapértelmezésben a **Sebesség** mértékegysége **km/h**, de ez felülírható az „**Options.UnitOfSpeed**” parancs használatával. A maximális sebesség használatához, javasolt egy magas sebességértéket beírni (például 9999 km/h), közvetlenül ugyanis nem lehet megadni „maximális sebességet”.

A parancs használatával a különböző jelzéképekhez különböző engedélyezett sebességet lehet beállítani. Amennyiben a jármű az engedélyezett sebességnél gyorsabban halad, akkor a játék során egy figyelemezhető üzenet utasítja a játékost, hogy csökkentse a sebességet.



A Japánban használt értékek az alapértelmezettek, amelyek a következők:

Index	Jelzési kép	Alapértelmezett sebesség
0	●	0 km/h
1	●●	25 km/h
2	●	55 km/h
3	●●	75 km/h
4	●	maximális
5	●●	maximális

[Példa a Route.Signal, Track.SigF és Track.Section használatára](#)

### Route.RunInterval

**Route.RunInterval** *ÉrtékMásodperc*

**ÉrtékMásodperc:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely vonatok követési távolságát jelenti **másodpercben**. Egy jármű közlekedik előttünk az itt beállított követési távolságban.

Ha ez a parancs nincs alkalmazva, akkor nincs előttünk haladó jármű.

A **Route.RunInterval** parancs megegyezik a [Train.Interval](#) parancssal.

### Route.AccelerationDueToGravity

**Route.AccelerationDueToGravity** *Érték*

**Érték:** Egy nem negatív szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a gravitáció gyorsulását jelenti, **m/s<sup>2</sup>**-ben (méter per szekundumnégzet). Az alapértelmezett érték **9.80665**.

Egyenlőnek, vagy nagyobbak kell lennie 0.01-nél.

Ez a beállítás hatással van a légnyomásra, légellenállásra, és sok más egyéb tényezőre.

### Route.Elevation

**Route.Elevation** *Magasság*

**Magasság:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a tengerszint feletti magasságot jelenti, a route fájl **0. szelvényénél**. Az alapértelmezett érték **0 méter**. Az érték függ az [Options.UnitOfLength](#) parancs beállításától.

Ez a beállítás hatással van a hőmérsékletre, légsűrűsége, légnyomásra, légellenállásra, és a sebesség hanghatásaira.

### Route.Temperature

**Route.Temperature** *ÉrtékCelsius*

**ÉrtékCelsius:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a levegő hőmérsékletét jelenti **Celsiusban**, a route fájl **0. szelvényénél**. Az alapértelmezett érték **20**.

Egyenlőnek, vagy nagyobbak kell lennie **1 Kelvinnél**, azaz **-272.15 Celsiusnál**.

Ez a beállítás hatással van a légsűrűsége, légnyomásra, légellenállásra, és a sebesség hanghatásaira.

### Route.Pressure

**Route.Pressure** *ÉrtékKPa*

**ÉrtékKPa:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a légnyomást jelenti **kPa-ban** (KiloPascal = 1000 Pascal), a route fájl **0. szelvényénél**. Az alapértelmezett érték **101.325**.

Egyenlőnek, vagy nagyobbak kell lennie 0.001-nél (1 Pascal).

Ez a beállítás hatással van a légnyomásra.



## Route.AmbientLight

### Route.AmbientLight Vörös; Zöld; Kék

A parancs az összes poligont megvilágítja a **Vörös, Zöld, Kék** összetevőkhöz használt színeknek megfelelően, figyelmen kívül hagyva a LightDirection parancsot.

**Vörös:** Egy egész szám **0 és 255 között**, amely a megvilágításhoz használt szín **vörös** összetevő száma. Az alapértelmezett érték **160**.

**Zöld:** Egy egész szám **0 és 255 között**, amely a megvilágításhoz használt szín **zöld** összetevő száma. Az alapértelmezett érték **160**.

**Kék:** Egy egész szám **0 és 255 között**, amely a megvilágításhoz használt szín **kék** összetevő száma. Az alapértelmezett érték **160**.

## Route.DirectionallLight

### Route.DirectionallLight Vörös; Zöld; Kék

A parancs megvilágítja a **Vörös, Zöld, Kék** összetevőkhöz használt színeknek megfelelően az összes poligon azon oldalait, amely irányból a fény érkezik. Az objektumoknak nem világítja meg azon részeit, amelyet nem ér fény. A [Route.LightDirection](#) parancssal lehet meghatározni a fény vetületének irányát (illetve a fényforrás irányát).

**Vörös:** Egy egész szám **0 és 255 között**, amely a megvilágításhoz használt szín **vörös** összetevő száma. Az alapértelmezett érték **160**.

**Zöld:** Egy egész szám **0 és 255 között**, amely a megvilágításhoz használt szín **zöld** összetevő száma. Az alapértelmezett érték **160**.

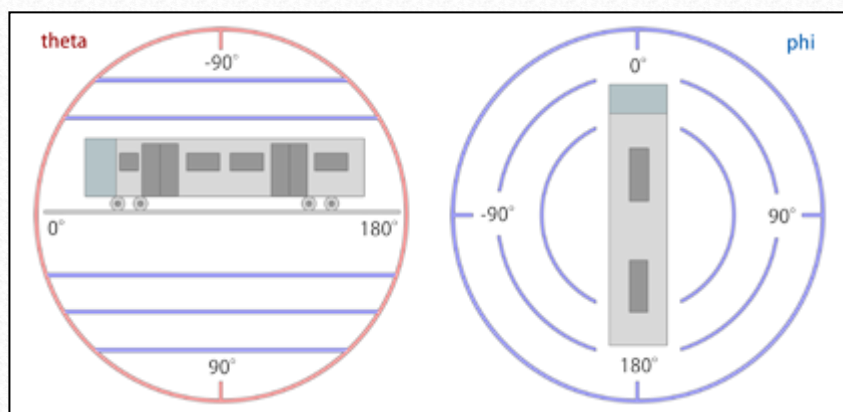
**Kék:** Egy egész szám **0 és 255 között**, amely a megvilágításhoz használt szín **kék** összetevő száma. Az alapértelmezett érték **160**.

## Route.LightDirection

### Route.LightDirection Theta; Phi

**Theta:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a fény vetületének szögét jelenti a vízszinteshez képest, **fokban**. Az alapértelmezett érték **60**.

**Phi:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a fény vetületének szögét jelenti az előre irányhoz képest, **fokban**. Az alapértelmezett érték **-26.57**.



A parancs a kezdeti fény vetületének szögét állítja be a **0. szelvényszámhoz**. Ezt jelenti azt, hogy a fény milyen szögben éri az objektumokat, illetve ennek az ellentétes irányában található a Nap (a fény forrása).

A **Theta** határozza meg a szöget a vízszinteshez képest. A **90** érték függőleges irányt jelent **entről lefelé**, a **-90** érték is függőleges irányt jelent, de **entről felfelé**. Ezeknél a szélsőséges értékeknél a **Phi** értéke lényegtelen. A **Theta 0** értéke egy irányt jelent, ami a horizontnál kezdődik. A magasságot a **Theta**, a forgást a síkban a **Phi**

értéke határozza meg. A **0** érték nem forog, a **90** érték az irányt jobbra, a **-90** érték az irányt balra forgatja. Hátulról előrefelé vetülő fényt be lehet állítani úgy, hogy a **Theta és Phi** értéke **180 és 0**, vagy az értékük **0 és 180**.

Közberső értékek használatával pontosabban be lehet állítani a fény vetületét (forrását).

## Route.DeveloperID

### Route.DeveloperID Szöveg

Ez a parancs az openBVE-ben nem használatos.

## „Train” szakasz

Ebben a szakaszban alkalmazható parancsok a [route fájlhoz](#) rendelt járműre vonatkoznak.

### Train.Folder; Train.File

**Train.Folder** *Mappanév*

**Train.File** *Mappanév*

**Mappanév:** A route fájlhoz rendelt jármű mappaneve.

### Train.Run; Train.Rail

**Train.Run**(*SíntípusIndex*) .Set *RunHangIndex*

**Train.Rail**(*SíntípusIndex*) .Set *RunHangIndex*

A .Set utótag nem kötelező.

**SíntípusIndex:** Egy nem negatív szám, amely egy, a [Structure.Rail](#) szakaszban már definiált síntípust jelent. Ezeket a síntípusokat a Track.Railtype paranccsal alkalmazzuk.

**RunHangIndex:** Egy nem negatív szám, amely a síntípushoz rendelt jármű **futáshangját** jelenti. A jármű mappájában található **RunHangIndex.wav** hangfájlt fogja a program lejátszani, amikor a jármű ezen a síntípuson halad.

### Train.Flange

**Train.Flange**(*SíntípusIndex*) .Set *FlangeHangIndex*

A .Set utótag nem kötelező.

**SíntípusIndex:** Egy nem negatív szám, amely egy, a [Structure.Rail](#) szakaszban már definiált síntípust jelent. Ezeket a síntípusokat a Track.Railtype paranccsal alkalmazzuk.

**FlangeHangIndex:** Egy nem negatív szám, amely a síntípushoz rendelt jármű **nyomkarima hangját** jelenti. A jármű mappájában található **RunHangIndex.wav** hangfájlt fogja a program lejátszani, amikor a jármű ezen a síntípuson halad.

### Train.Timetable (Day)

**Train.Timetable**(*MenetrendIndex*) .Day *Fájlnév*

A .Day utótag kötelező!

**MenetrendIndex:** A menetrend indexszáma (nem negatív szám).

**Fájlnév:** Az indexszámhoz rendelt **nappali** verziójú menetrend képfájl neve, a **Train** mappához viszonyított relatív útvonalon megadva.

A parancs a **MenetrendIndexhez** definiál egy **nappali** verziójú menetrendet. Ez a menetrend azon az állomáson jelenik meg az ajtók kinyitásakor, amelyek [Track.Sta](#) parancsban az adott **MenetrendIndex** be van állítva.

### Train.Timetable (Night)

**Train.Timetable**(*MenetrendIndex*) .Night *Fájlnév*

A .Night utótag kötelező!

**MenetrendIndex:** A menetrend indexszáma (nem negatív szám).

**Fájlnév:** Az indexszámhoz rendelt **éjszakai** verziójú menetrend képfájl neve, a **Train** mappához viszonyított relatív útvonalon megadva.

A parancs a **MenetrendIndexhez** definiál egy **éjszakai** verziójú menetrendet. Ez a menetrend azon az állomáson jelenik meg az ajtók kinyitásakor, amelyek [Track.Sta](#) parancsban az adott **MenetrendIndex** be van állítva.

---

**Train.Gauge****Train.Gauge** *ÉrtékMilliméter*

**ÉrtékMilliméter:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a route fájlban használt [nyomtávot](#) jelenti **milliméterben**. Az alapértelmezett érték **1435**.

**A Train.Gauge parancs megegyezik a [Route.Gauge](#) parancssal.**

---

**Train.Interval****Train.Interval** *ÉrtékMásodperc*

**ÉrtékMásodperc:** Egy szám (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely vonatok követési távolságát jelenti **másodpercben**. Egy jármű közlekedik előttünk az itt beállított követési távolságban.

Ha ez a parancs nincs alkalmazva, akkor nincs előttünk haladó jármű.

**A Train.Interval parancs megegyezik a [Route.RunInterval](#) parancssal.**

## „Structure” szakasz

A Structure szakaszban használt parancsok definiálják azokat az objektumokat, amelyek a későbbi parancsokban használhatóak. Például a **Track.Rail**, **Track.Freeobj**, stb parancsokban lévő **Indexszámmal** helyezhetők el azok az objektumok, amelyeket ebben a szakaszban egy **Indexszám** azonosít.

Figyelembe kell venni, hogy a **Freeobj** kivételével valamennyi objektum csak egy **blokk** kezdetén alkalmazható. A parancsok „**Track**” szakaszban található részletes leírásaiban külön figyelmeztetés van erre.

**Structure.Parancsnév(Index).Load Fájlnév**

A **.Load** utótag nem kötelező.

**Index:** Egy nem negatív egész szám, ami az adott objektumot fogja azonosítani (indexszám).

**Fájlnév:** A fájl neve, az **Object** mappához viszonyított **relatív útvonalat** kell megadni.

**Parancsnév:** Az alábbi parancsok közül az egyik:

Parancsnév:	Leírás
Rail	Objektumot definiál a <b>Track.Rail</b> , <b>Track.RailStart</b> és a <b>Track.RailType</b> parancsokhoz.
Beacon	Objektumot definiál a <b>Track.Beacon</b> parancshoz.
Ground	Objektumot definiál a <b>Track.Ground</b> parancshoz.
WallL	<b>Bal oldali</b> objektumot definiál a <b>Track.Wall</b> parancshoz.
WallR	<b>Jobb oldali</b> objektumot definiál a <b>Track.Wall</b> parancshoz.
DikeL	<b>Bal oldali</b> objektumot definiál a <b>Track.Dike</b> parancshoz.
DikeR	<b>Jobb oldali</b> objektumot definiál a <b>Track.Dike</b> parancshoz.
FormL	<b>Bal oldali</b> objektumot definiál a <b>Track.Form</b> parancshoz. Peronok pereméhez lehet használni.
FormR	<b>Jobb oldali</b> objektumot definiál a <b>Track.Form</b> parancshoz. Peronok pereméhez lehet használni.
FormCL	<b>Bal oldali</b> objektumot definiál a <b>Track.Form</b> parancshoz. Peronok közepéhez lehet használni. <b>„ANIMATED” kiterjesztésű fájl nem használható!</b>
FormCR	<b>Jobb oldali</b> objektumot definiál a <b>Track.Form</b> parancshoz. Peronok közepéhez lehet használni. <b>„ANIMATED” kiterjesztésű fájl nem használható!</b>
RoofL	<b>Bal oldali</b> objektumot definiál a <b>Track.Roof</b> parancshoz. Peronok peremének a tetejéhez lehet használni.
RoofR	<b>Jobb oldali</b> objektumot definiál a <b>Track.Roof</b> parancshoz. Peronok peremének a tetejéhez lehet használni.
RoofCL	<b>Bal oldali</b> objektumot definiál a <b>Track.Roof</b> parancshoz. Peronok közepének a tetejéhez lehet használni. <b>„ANIMATED” kiterjesztésű fájl nem használható!</b>
RoofCR	<b>Jobb oldali</b> objektumot definiál a <b>Track.Roof</b> parancshoz. Peronok közepének a tetejéhez lehet használni. <b>„ANIMATED” kiterjesztésű fájl nem használható!</b>
CrackL	<b>Bal oldali</b> objektumot definiál a <b>Track.Crack</b> parancshoz. <b>„ANIMATED” kiterjesztésű fájl nem használható!</b>
CrackR	<b>Jobb oldali</b> objektumot definiál a <b>Track.Crack</b> parancshoz. <b>„ANIMATED” kiterjesztésű fájl nem használható!</b>
FreeObj	Objektumot definiál a <b>Track.FreeObj</b> parancshoz.

Általában B3D, CSV, X és ANIMATED kiterjesztésű objektumfájlokat támogat a program. A jelzett objektumoknál azonban az ANIMATED kiterjesztésű objektumok nem támogatottak.



## Structure.Pole

**Structure.Pole**(VágányokSzama; OszlopIndex).Load *FájlNév*

A **.Load** utótag nem kötelező.

A paranccsal a felsővezeték tartó oszlopok különböző variációit lehet elkészíteni aszerint, hogy hány vágány felett jelenjenek meg. A [Track.Pole](#) parancs alkalmazásakor a **VágányokSzama** paraméter jelenti azt, hogy hány vágány felett jelenjen meg felsővezeték tartó oszlop.

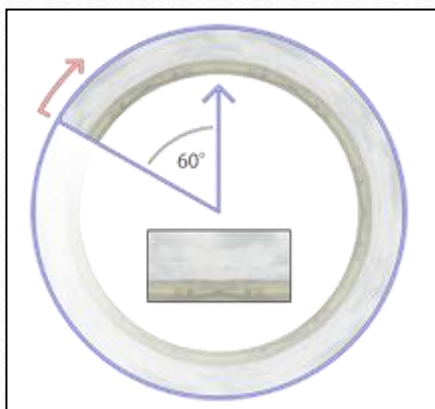
**VágányokSzama:** Egy nem negatív szám, amely azt jelenti, hogy hány vágány felett legyen felsővezeték tartó oszlop. A **0 egy vágányra**, az **1 két vágányra** készít felsővezeték tartót, stb.

**OszlopIndex:** Egy szám, amely az Oszloptípust fogja jelteni.

**FájlNév:** Az oszlopokhoz használt fájl neve, az **OszlopIndex** ezt az objektumot fogja azonosítani.

## „Texture” szakasz

Ebben a szakaszban lehet definiálni, hogy milyen háttérképeket lehessen a pályában használni, és ezek milyen módon jelenjenek meg.



A háttérkép (fentről nézve) egy „hengeres falként” jelenik meg a kamera körül, amelynek a kezdőpontja az **előre irányhoz képest 60 fokkal balra található** (a 10 óra irányában). Ebből a pontból kiindulva a henger körül egy képzeletbeli óralap van, amelyen a háttérkép textúra ismétlődik a **Texture.Background**(*BackgroundTextureIndex*).**X** parancsban beállított **Darabszámnak** megfelelően, melynek alapértelmezett értéke **egy teljes körben 6 darab**.

A kép felső 3/4-ed része a horizont felett, az alsó 1/4-ed része pedig a horizont alatt jelenik meg.

A **Texture.Background**(*BackgroundTextureIndex*).**Aspect** parancs használatával választani lehet, hogy a háttérképnek fix magassága legyen (0), vagy megtartsa az eredeti képméretarányát (1). Ha a háttérképnek fix magassága van, akkor a

henger magassága egyenlő a sugara felével, ami azt jelenti, hogy körülbelül 37 fokban hajlik a kép felső, és -14 fokban a kép alsó része. Ha a háttérkép megtartja az eredeti képméretarányt, akkor ez számításba veszi a kép magasságán és szélességén túl a **Darabszámot** is.

A **Darabszámra** való tekintet nélkül ügyelni kell arra, hogy a háttérkép bal és jobb széle illeszkedjen egymáshoz. Számításba kell még venni, hogy „sapkák” jönnek létre a háttérkép alsó és felső 10%-áról.

A pálya kezdetén a **Texture.Background**(0) parancshoz definiált háttérkép jelenik meg, ezt követően a [Track.Back](#) parancs alkalmazásával lehet a háttérképet változtatni.

## Texture.Background

**Texture.Background**(*HáttérképIndex*).Load *FájlNév*

A **.Load** utótag nem kötelező.

**FájlNév:** A háttérkép fájl neve, az **Object** mappához viszonyított **relatív útvonalat** kell megadni.

A parancs egy háttérképet definiál, amelyet később a [Track.Back](#) parancs alkalmazásával lehet használni vagy változtatni.

**Texture.Background**(*HáttérképIndex*).**X** *Darabszám*

Az **.X** utótag kötelező!

**Darabszám:** Az a szám, ahányszor a háttérkép ismétlődik egy teljes körben. Az alapértelmezett érték **6**.

**Texture.Background**(*HáttérképIndex*).**Aspect** *Mód*

Az **.Aspect** utótag kötelező!

**Mód:** A háttérkép megjelenítésének módja. Az alapértelmezett érték **0**.

**0:** A háttérképnek fix magassága legyen.

**1:** A háttérkép tartsa meg az eredeti képméretarányát.



## „Cycle” szakasz

### Cycle.Ground

```
Cycle.Ground(TalajIndex).Params TalajIndex0; TalajIndex1; ...; TalajIndexn-1
```

A **.Params** utótag nem kötelező.

**TalajIndex:** Egy nem negatív szám, amely egy definiált **Talajobjektumot** jelent, amely ciklusonként ismétlődik.

**TalajIndex<sub>indexszám</sub>:** Egy nem negatív szám, amely egy, a [Structure szakaszban](#) definiált **Ground (talaj) objektumot** jelent.

A [Track.Ground](#) parancs használatával ugyanaz a talajobjektum ismétlődik minden [blokkban](#). A Cycle.Ground alkalmazásával lehetőség van ezt felülírni, és sorozatban ismétlődő talajobjektumokat elhelyezni. A talajobjektumok így a megadott sorrend szerint ismétlődnek.

A következő két példa ugyanazt a végeredményt adja:

```
With Track
0, .Ground 0
25, .Ground 1
50, .Ground 2
75, .Ground 0
100, .Ground 1
125, .Ground 2
; stb...
```

Valamint:

```
With Cycle
.Ground(0) 0; 1; 2
With Track
0, .Ground 0
```

A 0. [szelvénytáblához](#) a **TalajIndex<sub>0</sub>** objektum kerül.

Az **sz** jelű szelvényhez a következő képlet szerint az **i** indexszámú **Talajindex** objektum kerül:

**maradék (sz / blokkhossz, n)**

Azaz a szelvénytábla és a blokkhossz hányadosa, és az objektumok számának hányadosának maradékával megegyező indexszámú tábla kerül az **sz** szelvényhez.

*Példa:*

Készítünk 3 féle talajobjektumot és definiáljuk a [Structure szakaszban](#):

```
With Structure
.Ground(3) fu.csv
.Ground(4) homok.csv
.Ground(5) beton.csv
```

A Cycle szakaszban sorrendben a 3, 4, 5 indexszámú objektumokat definiáljuk:

```
With Cycle
.Ground(0) 3; 4; 5
```

Így tehát **3 darab** objektumot definiáltunk a **Cycle.Ground(0)** indexszámhoz. A pálya [blokkhosszúsága 25 méter](#).

Ha ezt az objektumot a **575. szelvénytáblához** tesszük le, akkor a következő számítással kiszámolható, hogy melyik objektum fog megjelenni ebben a szelvényben:

```
With Track
575, .Ground 0
```

1. Szelvénytábla osztva a blokkhosszal:  $575 / 25 = 23$ .
2. Ezen érték osztva a darabszámmal, és lefelé kerekítve:  $23 / 3 = 7,6666666$ , lefelé kerekítve **7**.
3. A szelvénytábla és a blokkhossz hányadosából kivonva a darabszám és az előbbi érték szorzatát:
4.  $23 - 3 * 7 = 23 - 21 = 2$ .

Így az **575. szelvénytáblánál** a **Cycle.Ground** szakaszban álló **2-es indexszámú**, azaz a példa szerint a **Structure.Ground** szakaszban, a **5-ös indexszámúként** definiált **beton.csv** objektum fog megjelenni. Ezt követően a 600. szelvénynél a **Structure.Ground** szakaszban **3-asként** definiált **fu.csv**, a 625. szelvénynél a **Structure.Ground** szakaszban **4-esként** definiált **homok.csv** objektum jelenik meg, majd ezek sorrendben ismétlődnek.

## „Signal” szakasz

A paranccsal egyedi készítésű fényjelzőket lehet a pályába helyezni.

**Signal(TípusIndex).Load AnimáltObjektumFájl**

A **.Load** utótag nem kötelező.

**TípusIndex:** Egy nem negatív szám, amely a fényjelző típusát jelenti.

**AnimáltObjektumFájl:** A fényjelzőhöz használt animált objektum fájl neve, az **Object** mappához viszonyított útvonalon.

Ezt a parancsot lehet használni, ha animált fényjelzőt szeretnénk a pályában használni. A **TípusIndex** később a [Track.SigF](#) parancsban használható fel.

**Signal(TípusIndex).Load FényjelzőFájlKiterjesztésNélkül; IzzóFájlKiterjesztésNélkül**

A **.Load** utótag nem kötelező.

**TípusIndex:** Egy nem negatív szám, amely a fényjelző típusát jelenti.

**FényjelzőFájlKiterjesztésNélkül:** A fényjelzőhöz használt fájl neve, az **Object** mappához viszonyított útvonalon, de a fájl kiterjesztése nélkül megadva. **Kötelező megadni!**

**IzzóFájlKiterjesztésNélkül:** Az izzó képhez használt fájl neve, az **Object** mappához viszonyított útvonalon, de a fájl kiterjesztése nélkül megadva.

Ezt a parancsot lehet használni, ha egyedi készítésű textúrákkal használt fényjelzőket szeretnénk a pályába tenni. Az openBVE **B3D**, **CSV** vagy **X** kiterjesztésű fájlokat keres a megadott fájlneveken. A fényjelző és az izzó képek textúráinak elnevezésének az objektumfájlhoz képest ugyanolyan nevűnek kell lenniük, egy nem negatív számot kell hozzáírni, amely a fényjelző jelzési képét jelenti, és ezt követően a fájl kiterjesztése következik.

A **TípusIndex** számot később a [Track.SigF](#) parancsban lehet felhasználni.

Például a **FényjelzőFájlKiterjesztésNélkül** paraméter a következő fájlokat jelenti:

SignalFileWithoutExtension.**b3d**  
SignalFileWithoutExtension**0**.bmp  
SignalFileWithoutExtension**1**.bmp  
SignalFileWithoutExtension**2**.bmp  
SignalFileWithoutExtension<sub>n</sub>.bmp

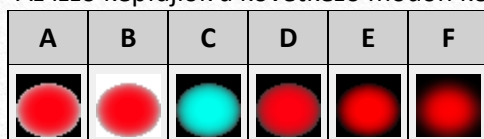
A jelzési képeket 0-tól kezdődően kell megadni, ahol **0** az **egy vörös fény**.

Az alapértelmezett jelzési képek: 0 (●), 1 (●●), 2 (●●), 3 (●●●), 4 (●●●) és 5 (●●●●).

A jelzési képeket a [Route.Signal](#) parancsban lehet beállítani tetszőlegesen.

Az objektumban mindig az a textúra kerül alkalmazásra, amely éppen aktív. Ez azt jelenti, hogy az objektumban nincs lehetőség más textúrákat használni. Az izzó képfájlokra ez szintén vonatkozik. Az izzó kép általában egy téglalap, amelyet a jelzés előtt helyezünk el, de különböző alakokat lehet használni.

Az izzó képfájlok a következő módon kerülnek feldolgozásra:



A kezdeti kép általában éles, és ovális alakú. A belseje teljesen tiszta, a külső szegélye pedig a fehér. A kép többi része tiszta fehér (A) vagy tiszta fekete (B) színű. Amikor az openBVE betölti az izzó képet, akkor a tiszta fekete pixelek tiszta fehér pixelekre cserélődnek, ennek az eredménye a „B” kép. Innen a kép inverz lesz (C), majd a színárnyalat megfordul (D). A „B” képpel összehasonlítva itt a képnek már világító hatása van, a telített pixelek változatlanok maradtak, de a fényesebb részek (a szegély) sötétebb lett. Innen a kép sötét részei még sötétebbek lesznek (E), majd kissé elkenődött lesz (F).

[Példa a Route.Signal, Track.SigF és Track.Section használatára](#)

## „Track” szakasz

Ebben a szakaszban állíthatjuk be a pálya tulajdonságait, vonalvezetését. Általában ez a szakasz található a route fájl végén.

A „Track” szakasz parancsaihoz [szelvényszámot](#) (távolságot) kell hozzárendelni. Ha egy szelvényszám megadásra került, akkor az ezt követő parancsok mindaddig ugyanebbe a szelvénybe kerülnek, amíg újabb szelvényszámot meg nem adunk. Ha az első parancs előtt nincs szelvényszám, akkor az automatikusan a 0. szelvényhez kerül. A szelvényszámokat (amikhez a parancsok tartoznak) nem szükséges növekvő sorrendben írni, mert a fájl betöltésénél ezek automatikusan növekvő sorrendbe rendeződnek. A szelvényszámoknál tizedesjegyek is alkalmazhatók (pontosan kell elválasztani a tizedesjegyeket), de néhány parancsot csak egy [blokk](#) kezdetén lehet elhelyezni. **Az alapértelmezett blokk hossza 25 méter**, így például ezek a parancsok csak a 25-tel osztható (0, 25, 50, 75, 100, 125, stb.) szelvénybe kerülhetnek. Ezek a parancsok a leírásban külön jelölve vannak. A mértékegység az [Options.UnitOfLength](#) paranccsal állítható.

### Vágányok:

#### Track.Railstart

**Track.RailStart** *VágányIndex; X; Y; SíntípusIndex*

Elindítunk vele egy új **vágányt**.

[Váltó \(kitérő\)](#) készítésére is alkalmas (csak a struktúrát készíti el, nem pedig az objektumot).

**VágányIndex:** A vágány indexszáma (értéke nagyobb, mint 0).

**X:** **Vízszintes** távolság a 0. vágánytól (negatív esetén **balra**, pozitív esetén **jobbra**)

**Y:** **Függőleges** távolság a 0. vágánytól (negatív esetén **lefelé**, pozitív esetén **felfelé**)

**SíntípusIndex:** A síntípus indexszáma, amely a [Structure.Rail](#) szakaszban már definiálva van.

Amennyiben a vágány távolságát változtatni szeretnénk a 0. vágányhoz képest, akkor a **Track.Rail** parancsot kell használni. Alapértelmezésben az *X*, *Y* és a *Síntípus* értéke mindig a korábban használt *VágányIndex*hez használt paraméterekkel egyenlő **méterben**, ha korábban nem volt ilyen *VágányIndex* használatban, akkor 0.

A vágány befejezéséhez a **Track.Railend** parancsot kell használni.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.

#### Track.Rail

**Track.Rail** *VágányIndex; X; Y; SíntípusIndex*

Ezzel a paraméterrel tudjuk megváltoztatni egy **másik vágány** tulajdonságait.

**VágányIndex:** A vágány indexszáma (értéke nagyobb, mint 0).

**X:** **Vízszintes** távolság a 0. vágánytól (negatív esetén **balra**, pozitív esetén **jobbra**)

**Y:** **Függőleges** távolság a 0. vágánytól (negatív esetén **lefelé**, pozitív esetén **felfelé**)

**SíntípusIndex:** A síntípus indexszáma, amely a [Structure.Rail](#) szakaszban már definiálva van.

Alapértelmezésben az *X*, *Y* és a *Síntípus* értéke mindig a korábban használt *VágányIndex*hez használt paraméterekkel egyenlő **méterben**, ha korábban nem volt ilyen *VágányIndex* használatban, akkor 0.

Ha valamelyik blokkban nem szeretnénk a vágány tulajdonságait átállítani, akkor a parancsot nem kell használni, ilyenkor a korábban használt értékek szerint halad a másik vágány. Csak akkor kell használni, ha a másik vágány valamelyik tulajdonságát (*X*, *Y*, *Síntípus*) meg akarjuk változtatni.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.

#### Track.Railtype

**Track.RailType** *VágányIndex; SíntípusIndex*

Kicseréli a sín textúráját, formáját egy, a [„Structure” szakasz](#), [„Rail” indexeknél](#) már meglévő síntípusra. Például íveknél, vagy ha másmilyenre változik az alépítmény a pályán (például fa talpfák helyett beton keresztaljak)

**VágányIndex:** A vágány indexszáma (amelyik vágányon a sín típusát meg akarjuk változtatni). Az indexszám 0 (0. vágány), vagy csak olyan vágánynak az indexszáma lehet, amely a [Track.Railstart](#) paranccsal már el lett kezdve, és a [Track.Railend](#) paranccsal még nem lett befejezve.

**SíntípusIndex:** Az új síntípus, forma indexszáma, amely a [Structure.Rail](#) szakaszban már definiálva van.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.



## Track.Railend

**Track.RailEnd** *VágányIndex; X; Y*

Vágány befejezésére szolgál. Például, ha azt szeretnénk, hogy az eddig kétvágányú pályából egyvágányú legyen.

**VágányIndex:** Annak a sínnek az indexszáma (nagyobb, mint 0), amit meg kívánunk szüntetni ebben a pontban. Csak olyan vágálynak az indexszáma lehet, amely a [Track.Railstart](#) paranccsal **már el lett kezdve**.

**X:** **Vízszintes** távolság a 0. vágánytól (negatív esetén **balra**, pozitív esetén **jobbra**).

**Y:** **Függőleges** távolság a 0. vágánytól **méterben** (negatív esetén **lefelé**, pozitív esetén **felfelé**)

Alapértelmezésben az X és Y értéke mindig az adott *VágányIndex*hez korábban használt paraméterekkel egyenlő **méterben**.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.

*Példa a RailStart, Railtype, Rail, és Railend parancsokra:*

```
With Track
1000, .RailStart 1; 4.2; 0.0; 0
1025, .RailType 1; 1
1050, .Rail 1; 3.8; 0.0; 0
1075, .RailEnd 1
```

*Az 1-es Indexszámú vágány a 0. vágánytól 4,2 méterre jobbra kezdődik, a 1000. szelvényzámnál.*

*1025. szelvénynél a vágány megtartja a korábban meghatározott 4,2 méteres távolságot, de a sín típusa*

*megváltozik az 1-es számú sín típusra. 1050. szelvénynél a vágány közelebb kerül a 0. vágányhoz, a távolsága 3,8*

*méter lesz, és újra a 0. indexszámú sín típus jelenik meg a vágányon. 1075. szelvényben az 1-es vágány befejeződik,*

*3,8 méterre a 0. vágánytól.*

## Track.Accuracy

**Track.Accuracy** *Érték*

A pálya állapotát állíthatjuk be vele. A vonat mozgása függ a beállítástól.

**Érték:** **1** esetén igen jól karbantartott pályát, **4** esetén elhanyagolt pályát jelent. Az alapértelmezett érték **2**.

## Track.Adhesion

**Track.Adhesion** *Érték*

A tapadás mértékét állíthatjuk be a paranccsal.

**Érték:** A tapadás mértéke. Az alapértelmezett érték **100**.

A száraz időre jellemző **Érték** a **135**, a hideg, fagyos időre jellemző **Érték** a **85**, a havas időre jellemző **Érték** az **50**. Amennyiben az **Érték** **0**, akkor a jármű mozgásképtelen.

## Vonalvezetés:

### Track.Curve

**Track.Curve** *Ívsugár; TúlemelésMilliméter*

Ívet lehet elhelyezni a pálya vonalába. Negatív **Ívsugár** esetén balra, pozitív **Ívsugár** eserén jobbra ível a pálya. **0** **Ívsugár** esetén a pálya egyenes vonalvezetésű.

**Ívsugár:** Az ív sugara. Az alapértelmezett érték **0 méter**.

**TúlemelésMilliméter:** Az ív túlemelésének mértéke **milliméterben**. Mindig az ívhez képesti belső sínszál lesz alacsonyabban, a külső sínszál lesz megemelve. Az alapértelmezett érték **0**.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.

*Megjegyzés: Túlemelés alkalmazásakor az objektumok nem változnak. Vezetés közben, illetve külső nézetben érzékelhető a képernyőn a jármű megdőlése. A valósághoz képest eltérő lehet a túlemelés dőlésszögének érzékelése, mert a valóságban a járművezető a járművel együtt megdől, szimulátor játék során a képernyő előtt ülő játékos viszont nem. [További információ a túlemelésről.](#)*

### Track.Pitch

**Track.Pitch** *ÉrtékEzrelék*

A pálya lejtésének, emelkedésének (függőleges vonalvezetés) változtatására alkalmas parancs.

**ÉrtékEzrelék:** A pálya lejtésének/emelkedésének mértéke **ezrelékben**.

Kiszámítása: a függőleges emelkedés 1000-szerese osztva a távolsággal (**ezrelék**).

$$\text{ÉrtékEzrelék} = 1000 * Y / X$$

1 ezreléknyi szintváltozás azt jelenti, hogy a pálya szintje 1000 méter hosszú szakaszon 1 métert változik függőlegesen.

**Pozitív** érték esetén emelkedés, **negatív** esetén lejtés. Amint ismét sík a terep, ott a **.Pitch 0**, parancsot kell használni.

Hosszabb blokk szakaszok esetén (**Options.BlockLength** magasabb), a nagyobb értékek változásakor a pályában hirtelen törés jelenhet meg.

Ez a parancs csak egy **blokk** kezdetén alkalmazható.

## Track.Turn

**Track.Turn** *Érték*

A parancs egy adott szelvényben megtöri a pályát a vízszintes irányú távolság (X) és a hosszanti irányú távolság hányadosa szerint.

**Érték:** vízszintes irányú távolság (X) osztva a hosszanti irányú távolsággal (Z).

$$\text{Érték} = X / Z$$

**Pozitív** esetén jobb irányú törés, **negatív** esetén bal irányú törés, **0** esetén egyenes.

Ez a parancs csak egy **blokk** kezdetén alkalmazható.

## Track.Height

**Track.Height** *Magasság*

A parancs azt állítja be, hogy a [0. vágány](#) milyen magasságban legyen a talajtól az adott [szelvényszám](#)ban.

**Magasság:** Egy **szám** (tizedesjegyek is alkalmazhatók, amelyet ponttal kell elválasztani), amely a 0. vágány magasságát jelenti a talaj felett, alapértelmezésben **méterben**.

Ez a parancs a [Structure.Ground](#) parancsban definiált talajobjektumok elhelyezését befolyásolja, amelyeket a [Track.Ground](#) alkalmazásával lehet változtatni. A magasság a két szomszédos **Track.Height** **Értéke** között lineáris átmenetként működik. Például a következő két példa egyenértékű:

1.: A két szomszédos Track.Height parancsok között fokozatos lineáris átmenet van:

```
1000, Track.Height 1
1075, Track.Height 4
```

2. A Track.Height parancsok minden 25 méter után új beállítást kapnak:

```
1000, Track.Height 1
1025, Track.Height 2
1050, Track.Height 3
1075, Track.Height 4
```

Ez a parancs csak egy **blokk** kezdetén alkalmazható.

## Objektumok:

### Track.Freeobj

**Track.Freeobj** *VágányIndex; ObjektumIndex; X; Y; ElforgatásY; ElforgatásX; ElforgatásZ; Térköz*

A parancs egy objektumot helyez el egy alkalommal.

**VágányIndex:** Annak a vágánynak az indexszáma, amelyhez viszonyítva az objektum elhelyezésre kerül. Az alapértelmezett érték **0**.

**ObjektumIndex:** Az objektum indexszáma, amely a **„Structure.Freeobj”** szakaszban már definiálva van. Az alapértelmezett érték **0**.

**X:** **Vízszintes** távolság a **VágányIndexben** meghatározott **egyenes vonalú vágánytól**. Negatív érték esetén balra, pozitív érték esetén jobbra. Az alapértelmezett érték **0**.

**Y:** **Függőleges** távolság a **VágányIndexben** meghatározott **egyenes vonalú vágánytól**. Negatív érték esetén a **sínkoronaszint** alatt, pozitív érték esetén felette. Az alapértelmezett érték **0**.

**ElforgatásY:** Az objektum elforgatásának szöge **fokban**, az **Y tengely** körül (azaz balra/jobbra). Másképpen: az elforgatás szöge az **XZ-síkot felülről nézve**, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.

**ElforgatásX:** Az objektum elforgatásának szöge **fokban**, az **X tengely** körül (azaz fel/le). Másképpen: az elforgatás szöge az **YZ-síkot balról nézve**, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.



**ElforgatásZ:**

Az objektum elforgatásának szöge **fokban**, a **Z tengely** körül (azaz az objektum hossz tengelyén nézve balra/jobbra). Másképpen: az elforgatás szöge az **XY-síkot hátulról** nézve, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.

**Track.Wall**

**Track.Wall** *VágányIndex; Oldal; FallIndex*

Folyamatos falat lehet elhelyezni a pálya mellett. A fal minden [blokkban](#) elhelyezésre kerül, egészen addig, amíg a [Track.Wallend](#) paranccsal nincs lezárva.

**VágányIndex:** A vágány indexszáma, amely mellé a falat elhelyezzük.

Az indexszám **0** (0. vágány), vagy csak olyan vágánynak az indexszáma lehet, amely a [Track.Railstart](#) paranccsal **már el lett kezdve**, és a [Track.Railend](#) paranccsal **még nem lett befejezve**.

**Oldal:** Ezzel állítható be, hogy a vágány melyik oldalára helyezzük a falat.

**-1** esetén a **WallL** objektum kerül elhelyezésre a vágány bal oldalára,

**0** esetén a **WallL** és **WallR** objektum is elhelyezésre kerül, a vágány mindkét oldalára.

**1** esetén a **WallR** objektum kerül elhelyezésre a vágány a jobb oldalra,

**FallIndex:** A fal indexszáma, amely a **Structure szakaszban** [„Structure.WallL”](#) vagy [„Structure.WallR”](#) objektumként már definiálva van.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.

**Track.Wallend**

**Track.Wallend** *VágányIndex*

A [Track.Wall](#) paranccsal már megkezdett falat fejez be.

**VágányIndex:** Annak a vágánynak az indexszáma, amelyikhez tartozó falat szeretnénk lezárni.

Az indexszám **0** (0. vágány), vagy csak olyan vágánynak az indexszáma lehet, amely a [Track.Railstart](#) paranccsal már el lett kezdve, és a [Track.Railend](#) paranccsal még nem lett befejezve, illetve amely vágányhoz már elkezdtünk egy falat a [Track.Wall](#) paranccsal.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.

**Track.Dike**

**Track.Dike** *VágányIndex; Oldal; TöltésIndex*

Folyamatos töltést lehet elhelyezni a pálya mellett. A töltés minden [blokkban](#) elhelyezésre kerül, egészen addig, amíg a [Track.Dikeend](#) paranccsal nincs lezárva.

**VágányIndex:** A vágány indexszáma, amely mellé a töltést elhelyezzük.

Az indexszám **0** (0. vágány), vagy csak olyan vágánynak az indexszáma lehet, amely a [Track.Railstart](#) paranccsal **már el lett kezdve**, és a [Track.Railend](#) paranccsal **még nem lett befejezve**.

**Oldal:** Ezzel állítható be, hogy a vágány melyik oldalára helyezzük a töltést.

**-1** esetén a **DikeL** objektum kerül elhelyezésre a vágány bal oldalára,

**0** esetén a **DikeL** és **DikeR** objektum is elhelyezésre kerül, a vágány mindkét oldalára.

**1** esetén a **DikeR** objektum kerül elhelyezésre a vágány a jobb oldalra,

**TöltésIndex:** A fal indexszáma, amely a **Structure szakaszban** [„Structure.DikeL”](#) vagy [„Structure.DikeR”](#) objektumként már definiálva van.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.

**Track.Dikeend**

**Track.Dikeend** *VágányIndex*

A [Track.Dike](#) paranccsal már megkezdett töltést fejez be.

**VágányIndex:** Annak a vágánynak az indexszáma, amelyikhez tartozó töltést szeretnénk lezárni.

Az indexszám **0** (0. vágány), vagy csak olyan vágánynak az indexszáma lehet, amely a [Track.Railstart](#) paranccsal már el lett kezdve, és a [Track.Railend](#) paranccsal még nem lett befejezve, illetve amely vágányhoz már elkezdtünk egy töltést a [Track.Dike](#) paranccsal.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.

**Track.Pole****Track.Pole** *VágányIndex; VágányokSzama; Oldal; Intervallum; OszlopIndex*

Felsővezeték tartó oszlopokat lehet elhelyezni a vágányok mentén.

**VágányIndex:** A vágány indexszáma, amely mellé a felsővezeték tartó oszlop elhelyezésre kerül.Az indexszám **0** (0. vágány), vagy csak olyan vágálynak az indexszáma lehet, amely a [Track.Railstart](#) paranccsal már el lett kezdve, és a [Track.Railend](#) paranccsal még nem lett befejezve.**VágányokSzama:** Azon vágányok száma mínusz egy, ahányak szeretnénk ha lenne felsővezetéke.Példuál **2** vágány esetén a **VágányokSzama 1**.**Oldal:** Az oszlop helye. **-1** esetén bal oldalon, **1** esetén jobb oldalon.**Intervallum:** Két oszlop között lévő távolság. Példuál **25** esetén **25 méterenként** ismétlődnek az oszlopok.**OszlopIndex:** A fal indexszáma, amely a „[Structure.Pole](#)” szakaszban már meglévő felsővezeték tartó oszlop indexszáma.**Track.Poleend****Track.Poleend** *VágányIndex*

A paranccsal már elkezdett felsővezeték tartó oszlopokat lehet befejezni.

**VágányIndex:** Annak a vágálynak az indexszáma, amelyikhez tartozó felsővezeték tartó oszlopot szeretnénk lezárni.Az indexszám **0** (0. vágány), vagy csak olyan vágálynak az indexszáma lehet, amely a [Track.Railstart](#) paranccsal már el lett kezdve, és a [Track.Railend](#) paranccsal még nem lett befejezve, illetve amely vágányhoz már elkezdtünk egy falat a [Track.Pole](#) paranccsal.Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.**Track.Crack****Track.Crack** *VágányIndex<sub>1</sub>; VágányIndex<sub>2</sub>; KitöltésIndex*

Nincs leírás.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.**Track.Ground**

Nincs leírás.

Ez a parancs csak egy [blokk](#) kezdetén alkalmazható.**Állomások:****Track.Sta****Track.Sta** *ÁllomásNév; ÉrkezésiIdő; IndulásiIdő; ÁthaladásFigyelmeztetés; AjtóOldal; Kijáratijelző; BiztonságiRendszer; ÉrkezésHang; TartózkodásiIdő; Utasmennyiség; IndításjelzőHang; MenetrendIndex*Ez a parancs egy állomást helyez el a pályán. A parancsot javasolt a peron elejéhez beírni. A [Track.Stop](#) parancs használatával lehet a járművek megállási helyét elhelyezni. Az ezt követő [Track.Stop](#) parancsok az adott állomáshoz tartoznak. Legalább egy [Track.Stop](#) parancsot kötelező elhelyezni.**Állomásnév:** Az állomás neve. Ez a szöveg jelenik meg a menetrendben, illetve az üzenetekben, így a javasolt valamilyen szöveget beírni ide.**ÉrkezésiIdő:** A játékos által vezetett jármű pontos érkezési időpontja az adott állomásra. Az alapértelmezett érték **0**.**Beállítási lehetőségek:****Időpont:** A jármű pontos érkezési időpontja (**óó.pppmm** formátumban!).**Üresen hagyva:** A jármű bármilyen időpontban érkezhetsz.**P:** A jármű nem áll meg ezen az állomáson, hanem megállás nélkül áthalad.**IndulásiIdő:** A játékos által vezetett jármű pontos indulási időpontja az adott állomásról. Az alapértelmezett érték **0**.**Beállítási lehetőségek:****Időpont:** A jármű pontos indulási időpontja (**óó.pppmm** formátumban!).**Üresen hagyva:** A jármű bármilyen időpontban érkezhetsz.

**T:** Ez az állomás a vonal végállomása, a járműnek nincs indulási időpontja.

**ÁthaladásFigyelmeztetés:** Az áthaladásra figyelmeztető berendezés állapota, amely arra figyelmezteti a jármű vezetőjét, hogy az állomáson meg kell-e állnia. Az alapértelmezett érték **0**.

**Beállítási lehetőségek:**

**0:** Az áthaladásra figyelmeztető berendezés **nem figyelmezteti** a járművezetőt az állomásban történő megállásról.

**1:** Az áthaladásra figyelmeztető berendezés **figyelmezteti** a járművezetőt az állomásban történő megállásról.

**AjtóOldal:** Meghatározza, hogy a jármű mely oldalán legyenek nyithatóak az ajtók az állomásban. Az alapértelmezett érték **0**.

**Beállítási lehetőségek:**

**L vagy -1:** A jármű **bal oldali** ajtajai nyithatóak.

**N vagy 0:** A jármű ajtajai **nem nyithatóak**.

**R vagy 1:** A jármű **jobb oldali** ajtajai nyithatóak.

**B:** A jármű **bal és jobb oldali** ajtajai egyszerre nyithatóak.

**Kijáratijelző:** Meghatározza, hogy az állomást követően a fényjelzőnek vörösnek, vagy továbbhaladást engedélyező állásban kell lennie. Az alapértelmezett érték **0**.

**Beállítási lehetőségek:**

**0:** A fényjelzők továbbhaladást engedélyező állásban vannak.

**1:** A fényjelzők vörös jelzést adnak a jármű megérkezéséig, illetve az indulási idő eléréskéig.

**BiztonságiRendszer:** Meghatározza, hogy a melyik biztonsági berendezés alkalmazható a következő állomásig.

**Beállítási lehetőségek:**

**ATS vagy 0:** Az **ATS** biztonsági berendezés alkalmazható.

**ATC vagy 1:** Az **ATC** biztonsági berendezés alkalmazható.

**ÉrkezésHang:** A **Sound** mappához viszonyított fájl lejátszódik az érkezéskor, az ajtók nyitását követően.

**Tartózkodásildő:** Az állomásban tartózkodás minimális ideje **másodpercben**, beleértve az ajtónyitás és zárás idejét is. Az alapértelmezett érték **15**.

**Utasmennyiség:** Egy pozitív szám, amely az állomásról történő induláskor a járművön utazó **utasmennyiséget** jelenti. A **100** érték egy normál utasmennyiséget jelent, a **250** érték pedig egy teljesen zsúfolt mennyiséget. **0 és 250 közötti érték** használható. Az alapértelmezett érték **100**.

**IndításjelzőHang:** A **Sound** mappához viszonyított fájl lejátszódik az indulás előtt (indulási idő mínusz hangfájl hossza, mínusz ajtózáras ideje).

**MenetrendIndex:** Egy nem negatív szám, amely azt jelenti, hogy a [Train.Timetable](#) szakaszban definiált menetrendek közül melyik jelenjen meg az állomásról történő indulást követően.

## Track.Station

**Track.Station** ÁllomásNév; Érkezésildő; Indulásildő; Kijáratijelző; BiztonságiRendszer; IndításjelzőHang

Ez a parancs egy állomást helyez el a pályán. **Javasolt a [Track.Sta](#) parancs használata a több beállítási lehetőség miatt.** A parancsot javasolt a peron elejéhez beírni. A [Track.Stop](#) parancs használatával lehet a járművek megállási helyét elhelyezni. Az ezt követő Track.Stop parancsok az adott állomáshoz tartoznak. Legalább egy Track.Stop parancsot kötelező elhelyezni.

**Állomásnév:** Az állomás neve. Ez a szöveg jelenik meg a menetrendben, illetve az üzenetekben, így a javasolt valamilyen szöveget beírni ide.

**Érkezésildő:** A játékos által vezetett jármű pontos érkezési időpontja az adott állomásra. Az alapértelmezett érték **0**.

**Beállítási lehetőségek:**

**Időpont:** A jármű pontos érkezési időpontja (**óó.pppmm** formátumban!).

**Üresen hagyva:** A jármű bármilyen időpontban érkezhet.

**P:** A jármű nem áll meg ezen az állomáson, hanem megállás nélkül áthalad.

**Indulásildő:** A játékos által vezetett jármű pontos indulási időpontja az adott állomásról. Az alapértelmezett érték **0**.

**Beállítási lehetőségek:**

**Időpont:** A jármű pontos indulási időpontja (**óó.pppmm** formátumban!).

**Üresen hagyva:** A jármű bármilyen időpontban érkezhetsz.

**T:** Ez az állomás a vonal végállomása, a járműnek nincs indulási időpontja.

**Kijáratijelző:** Meghatározza, hogy az állomást követően a fényjelzőnek vörösnek, vagy továbbhaladást engedélyező állásban kell lennie. Az alapértelmezett érték **0**.

**Beállítási lehetőségek:**

**0:** A fényjelzők továbbhaladást engedélyező állásban vannak.

**1:** A fényjelzők vörös jelzést adnak a jármű megérkezéséig, illetve az indulási idő eléréskéig.

**BiztonságiRendszer:** Meghatározza, hogy a melyik biztonsági berendezés alkalmazható a következő állomásig.

**Beállítási lehetőségek:**

**ATS vagy 0:** Az **ATS** biztonsági berendezés alkalmazható.

**ATC vagy 1:** Az **ATC** biztonsági berendezés alkalmazható.

**IndításjelzőHang:** A **Sound** mappához viszonyított fájl lejátszódik az indulás előtt (indulási idő mínusz hangfájl hossza, mínusz ajtózáras ideje).

A **Track.Sta** parancsban beállítható, de a **Track.Station** parancsban nem megadható paraméterekre a következő értékeket veszi figyelembe a program:

ÁthaladásFigyelmeztetés	0
AjtóOldal	B (mindkét oldal)
ÉrkezésHang	Nem játszódik le
Tartózkodásildő	15
Utasmennyiség	100
MenetrendIndex	Nem használt

**Track.Stop**

**Track.Stop** Oldal; HatárHátra; HatárElőre; KocsikSzáma

A parancs egy „**Megállás helye**” jelzőt helyez, ez jelöli az állomásokban azt a helyet, ahol a jármű elejével meg kell állni, a **KocsikSzáma** függvényében.

**Oldal:** Egy alapértelmezett „**Megállás helye**” jelző helyezhető el, a vágány valamelyik oldalára. **-1** esetén a vágány bal, **1** esetén a jobb oldalára. **0** esetén nem kerül elhelyezése, ilyenkor emelett egy **Track.Freeobj** parancs alkalmazásával helyettesíthetjük a jelzőt egy saját objektummal.

**HatárHátra:** Egy nem negatív szám, amely a „**Megállás helye**” jelzőtől számított távolságot jelenti **hátrafelé**. Ezen a távolságon belül kell a járműnek megállnia. Ha a jármű ezen a távolsághatáron kívül áll meg, akkor az ajtók nem nyithatóak. Ilyenkor figyelmeztető üzenet jelenik meg, a jármű helyének pontosítására.

**HatárElőre:** Egy nem negatív szám, amely a „**Megállás helye**” jelzőtől számított távolságot jelenti **előrefelé**. Ezen a távolságon belül kell a járműnek megállnia. Ha a jármű ezen a távolsághatáron kívül áll meg, akkor az ajtók nem nyithatóak. Ilyenkor figyelmeztető üzenet jelenik meg, a jármű helyének pontosítására.

**KocsikSzáma:** Egy pozitív egész szám, amely a szerelvénybe sorolt kocsik számát jelenti. Több **Track.Stop** parancs esetén annál a „**Megállás helye**” jelzőnél kell megállni, amelynél a **KocsikSzáma** paraméter megegyezik szerelvénybe sorolt kocsik számával. A **KocsikSzáma** paraméterbe minden egyes külön kezelhető járművet bele kell számolni (mozdony + kocsik).

**Példa a Track.Stop parancs alkalmazására**

**Track.Form**

**Track.Form** VágányIndex<sub>1</sub>; VágányIndex<sub>2</sub>; TetőIndex; PeronIndex

Nincs leírás.



**Jelzők:****Track.Limit****Track.Limit** *Sebesség; Oldal; Irány*

Ezzel a paranccsal a pályára engedélyezett sebességet változtathatjuk az adott ponttól.

**Sebesség:** Egy **szám**, amely az engedélyezett sebességet jelenti vagy 0, amely megszünteti a sebességkorlátozást. Az alapértelmezett érték **0 km/h**.**Oldal:** Az az **oldal**, amelyiken az alapértelmezett Japánban alkalmazott sebességkorlátozó tábla megjelenjen. Az alapértelmezett érték **0**.**-1** = A pálya **bal** oldalon elhelyezett sebességkorlátozó jelző.**0** = Sebességkorlátozó tábla **nem kerül elhelyezésre**, ezt saját objektummal is helyettesíthetjük, a **Freeobj** paranccsal.**1** = A pálya **jobb** oldalon elhelyezett sebességkorlátozó jelző.**Irány:** A beállított iránynak megfelelő irányú ívben lesz sebességkorlátozás. Az alapértelmezett érték **0**.**-1** = **Bal** irányú ívekben lesz sebességkorlátozás.**0** = Irány nincs meghatározva.**1** = **Jobb** irányú ívekben lesz sebességkorlátozás.

Példa:



Az alapértelmezett, Japánban alkalmazott sebességkorlátozó tábla

A jelentésük balról jobbra haladva sorrendben:

nincs sebességkorlátozás (.Limit 0; x)

Csak **bal** irányú ívekben **45 km/h** sebességkorlátozás, (.Limit 45; x; -1)45 km/h sebességkorlátozás, **iránytól függetlenül** (.Limit 45; x; 0)Csak **jobb** irányú ívekben **45 km/h** sebességkorlátozás, (.Limit 45; x; 1)Ha az új sebességkorlátozás értéke **alacsonyabb**, mint az aktuális engedélyezett sebesség, akkor az adott ponttól kezdve az új, alacsonyabb sebességkorlátozás az érvényes.Ha az új sebességkorlátozás értéke **magasabb**, mint az aktuális engedélyezett sebesség, akkor az új sebességkorlátozás csak akkor lesz érvényes, ha a vonat **teljes egészében lehaladt** a régi, alacsonyabb sebességre engedélyezett pályaszakasról.Ha a **Sebességet 0-ra** állítjuk, akkor a sebességkorlátozás megszűnik. Az Oldal **-1**, vagy **1**-re állításával a Japánban alkalmazott sebességkorlátozó tábla jelenik meg a pálya beállított oldalán.Az **Irány** beállításával csak a beállított iránynak megfelelő ívekben lesz sebességkorlátozás.**Track.Section****Track.Section** *Térköz<sub>0</sub>; Térköz<sub>1</sub>; ...; Térköz<sub>n</sub>*

A fényjelző mögötti foglalt és szabad térközökre vonatkozó jelzési képek beállítására alkalmas parancs.

**térköz<sub>x</sub>**: Egy **indexszám**, amely a fényjelző egyik jelzési képét jelenti.A parancs egy új térközt határoz meg, a **Route.Signal** és a **Track.SigF** parancsokkal **szoros összefüggésben van**. A **0 indexszám a vörös jelzőnek felel meg**, amely a járművel nem haladható meg. Az első paramétert kötelező megadni.Két különböző térközműködést lehet beállítani, **index-alapút** vagy **érték-alapút**. Ezt az **Options.SectionBehavior** paranccsal állíthatjuk be, az **alapértelmezett érték az index-alapú**.▪ **Index-alapú térközműködés:**A parancs paramétereiben azt kell megadni, hogy az adott fényjelzőn melyik indexszámú jelzési kép jelenjen meg attól függően, hogy az adott fényjelzőt követő hányadik jelző vörös. Egy jelző akkor vörös, ha az azt követő térköz foglalt, vagy az állomás kijárat szakaszán található jelző a **Track.Sta** parancsban vörösre van állítva.A **Route.Signal** parancsban az indexszámok jelzésképet is jelentenek, amelyhez sebességértéket rendelhetünk.



**térköz<sub>0</sub>:** Melyik **indexszámú** jelzési kép jelenjen meg a fényjelzőn, ha a **fényjelző mögötti jelző vörös**.

**térköz<sub>1</sub>:** Melyik **indexszámú** jelzési kép jelenjen meg a fényjelzőn, ha a fényjelző mögötti **2. jelző vörös**.

**térköz<sub>2</sub>:** Melyik **indexszámú** jelzési kép jelenjen meg a fényjelzőn, ha a fényjelző mögötti **3. jelző vörös**.

**térköz<sub>x</sub>:** Melyik **indexszámú** jelzési kép jelenjen meg a fényjelzőn, ha a fényjelző mögötti **x+1. jelző vörös**.

#### ▪ Érték-alapú térközműködés:

Az érték-alapú beállításnál a **Track.Section** parancs paramétereibe az adott fényjelző jelzéseképeit sorolhatjuk fel (a sorrendje lényegtelen), és ezen jelzéseképek közül a program a lehető legkisebb olyan jelzéseképet fogja használni, ami nagyobb a következő fényjelző jelzéseképénél.

Példa:

```
.Section 0;2;4
```

A példában ha a következő térköz jelzéseképe vörös (0), akkor az adott fényjelző jelzéseképe egy sárga fény lesz (2). Ha a következő térköz jelzéseképe egy sárga (2), akkor az adott jelző zöld lesz (4).

[Példa a Route.Signal, Track.SigF és Track.Section használatára](#)

### Track.SigF

**Track.SigF** *Típus; Térköz; X; Y; ElforgatásY; ElforgatásX; ElforgatásZ*

Egy működő fényjelzőt helyez el a pálya mentén, amelynek jelzéseire saját textúrákat is készíthetünk.

**Típus:** egy szám, amely a fényjelző típusát jelenti. A „Structure” szakasz **Signal** részében kell definiálni a jelzőtípusokat.

**Térköz:** működése ismeretlen. (1-es érték használatával rendben működik).

**X:** a jelző helye a 0. vágánytól **vízszintesen** negatív érték esetén **balra**, pozitív érték esetén **jobbra**. Az alapértelmezett érték **0 méter**.

**Y:** a jelző helye a 0. vágánytól számítva függőlegesen **felfelé**. Az alapértelmezett érték **0 méter**.

**ElforgatásY:** A jelző elforgatásának szöge **fokban**, az **Y tengely** körül (azaz balra/jobbra). Másképpen: az elforgatás szöge az **XZ-síkot felülről nézve**, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.

**ElforgatásX:** A jelző elforgatásának szöge **fokban**, az **X tengely** körül (azaz fel/le). Másképpen: az elforgatás szöge az **YZ-síkot balról nézve**, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.

**ElforgatásZ:** A jelző elforgatásának szöge **fokban**, a **Z tengely** körül (azaz az objektum hossz tengelyén nézve balra/jobbra). Másképpen: az elforgatás szöge az **XY-síkot hátulról nézve**, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.

[Példa a Route.Signal, Track.SigF és Track.Section használatára](#)

### Track.Signal

**Track.Signal** *Típus; Szöveg; X; Y*

**Track.Sig** *Típus; Szöveg; X; Y*

Fényjelző elhelyezésére alkalmas parancs.

**Típus:** egy szám, amely a fényjelző típusát jelenti. Az alapértelmezett érték **-2**. A következő típusok lehetségesek:

- 2 = Két** funkciós fényjelző (A típus), amelynek a jelzéseképei: ● egy vörös fény, ● egy sárga fény.
- 2 = Két** funkciós fényjelző (B típus), amelynek a jelzéseképei: ● egy vörös fény, ● egy zöld fény.
- 3 = Három** funkciós fényjelző, amelynek a jelzéseképei: ● egy vörös fény, ● egy sárga fény, ● egy zöld fény.
- 4 = Négy** funkciós fényjelző (A típus), amelynek a jelzéseképei: ● egy vörös fény, ●● két sárga fény ● egy sárga fény, ● egy zöld fény.
- 4 = Négy** funkciós fényjelző (B típus), amelynek a jelzéseképei: ● egy vörös fény, ● egy sárga fény, ●● sárga-zöld fény, ● egy zöld fény.
- 5 = Öt** funkciós fényjelző (A típus), amelynek a jelzéseképei: ● egy vörös fény, ●● két sárga fény, ● egy sárga fény, ●● sárga-zöld fény, ● egy zöld fény.
- 5 = Öt** funkciós fényjelző (B típus), amelynek a jelzéseképei: ● egy vörös fény, ● egy sárga fény, ●● sárga-zöld fény, ● egy zöld fény, ●● két zöld fény.
- 6 = Hat** funkciós fényjelző, amelynek a jelzéseképei: ● egy vörös fény, ●● két sárga fény, ● egy sárga fény, ●● sárga-zöld fény, ● egy zöld fény, ●● két zöld fény.

**Szöveg:** Ez a paraméter az openBVE-ben nem használatos.

**X:** a jelző helye a 0. vágánytól **vízszintesen** negatív érték esetén **balra**, pozitív érték esetén **jobbra**. Az alapértelmezett érték **0 méter**.

**Y:** a jelző helye a 0. vágánytól számítva függőlegesen **felfelé**. Az alapértelmezett érték **0 méter**.

Ha **X = 0**, akkor a jelző **működik**, azonban **láthatatlan**.

Ha **X** nem egyenlő 0-val és az **Y** **negatív** szám, akkor az **Y** értéke **4.8** értéket vesz fel, és egy alapértelmezett jelzőoszlopra kerül a fényjelző.

*Példa egy négy funkciós (B típus) fényjelző elhelyezésére a vágánytengelytől balra 3 méterre, felfelé 5 méterre, alapértelmezett jelzőoszlop nélkül:*

```
1000, Track.Signal -4;;-3;5
```

*Példa egy négy funkciós (B típus) fényjelző elhelyezésére a vágánytengelytől balra 3 méterre, felfelé 5 méterre, alapértelmezett jelzőoszloppal:*

```
1000, Track.Signal -4;;-3;-1
```

## Track.Relay

**Track.Relay X; Y**

A parancs egy alapértelmezett japán ismétlőjelzőt helyez el. Az ismétlőjelző a következő fényjelző jelzését mutatja.

**X:** **Vízszintes** távolság a 0. vágánytól (negatív esetén **balra**, pozitív esetén **jobbra**).

Az alapértelmezett érték **0 méter**.

**Y:** **Függőleges** távolság a 0. vágánytól (negatív esetén **lefelé**, pozitív esetén **felfelé**).

Az alapértelmezett érték **0 méter**.

Ha az **Y** értéke negatív, akkor az **Y** értéke 4.8 lesz, és az ismétlőjelző egy oszlopra kerül elhelyezésre.

## Biztonsági rendszerek:

### Track.Beacon

Nincs leírás.

### Track.Transponder

Nincs leírás.

### Track.AtsSn

Nincs leírás.

### Track.AtsP

Nincs leírás.

### Track.Pattern

Nincs leírás.

### Track.PLimit

Nincs leírás.

## Egyéb:

### Track.Back

**Track.Back HáttérképTextúraIndex**

A paranccsal a pálya háttérképét lehet megváltoztatni.

**HáttérképTextúraIndex:** A háttérkép indexszáma, amely a [„Texture.Background” szakaszban](#) már definiálva van. Ezt követően a pálya háttérképeként az új kép fog megjelenni.

Ez a parancs csak egy **blokk** kezdetén alkalmazható.

## Track.Fog

**Track.Fog** *TávolságKezdete; TávolságVége;Vörös; Zöld; Kék*

A paranccsal „kód effektet” lehet létrehozni, illetve megszüntetni.

**TávolságKezdete:** Egy nem negatív szám, amely „kód effekt” **távolságának kezdetét** jelenti, alapértelmezésben **méterben**. Az alapértelmezett érték **0**.

**TávolságVége:** Egy nem negatív szám, amely „kód effekt” **távolságának végét** jelenti, alapértelmezésben **méterben**. Az alapértelmezett érték **0**.

**Vörös:** Egy **egész szám 0 és 255 között**, amely a „kód effekthez” használt szín **vörös** összetevő száma. Az alapértelmezett érték **128**.

**Zöld:** Egy **egész szám 0 és 255 között**, amely a „kód effekthez” használt szín **zöld** összetevő száma. Az alapértelmezett érték **128**.

**Kék:** Egy **egész szám 0 és 255 között**, amely a „kód effekthez” használt szín **kék** összetevő száma. Az alapértelmezett érték **128**.

Kód effekt akkor jön létre, ha a **TávolságKezdete** kisebb, mint a **TávolságVége**, és a **TávolságKezdete** kevesebb 600 méternél. Mindkét értéknek nem negatívnak kell lennie.

Kód effektet akkor lehet megszüntetni, ha a **TávolságKezdete** nagyobb vagy egyenlő, mint a **TávolságVége** vagy a **TávolságKezdete** nagyobb vagy egyenlő 600 méternél.

A kód effekt színezi az objektumokat. A **TávolságKezdete** előtt látható objektumok változatlan formában jelennek meg, a **TávolságVége** után látható objektumok a kód effekt színe szerint lesznek láthatóak. A két távolság között lévő objektumok színe a kód színének megfelelően lineárisan változnak a távolság függvényében.

A háttérkép színe a kódnek megfelelő színű lesz. A program úgy tekinti, hogy a háttérkép 600 méterre van a kamerától, függetlenül a pillanatnyi nézőpont helyétől.

Két lehetőség van a kód effektek beállítására. Lásd a [Options.FogBehavior](#) parancs beállításait.

Ez a parancs csak egy **blokk** kezdetén alkalmazható.

## Track.Brightness

**Track.Brightness** *Érték*

Ez a parancs a környezet fényerejét befolyásolja.

**Érték:** Egy szám, amely a fényerő értékét jelenti. Alacsony érték sötét, magas érték világos fényerőt jelent.

A fényerő a két szomszédos **Track.Brightness** **Értéke** között lineáris átmenetként működik. A parancs hatással van a vezethető jármű külső és belső nézetére. Világosabb fényerő használata esetén az **nappali verziójú textúrák**, sötétebb fényerő használata esetén az **éjszakai verziójú textúrák** jelennek meg. Amennyiben **éjszakai verziójú textúra** nincs megadva, akkor a **nappali verziójú textúra** fényereje változik.

[Példa a Track.Brightness parancs alkalmazására](#)

## Track.Marker

**Track.Marker** *Fájlnev; Távolság*

Megjelenít egy képet a képernyő jobb felső részében.

**Fájlnev:** A megjelenítendő fájl neve, az **Object** mappához viszonyított **relatív útvonalat** kell megadni.

**Távolság:** Egy szám, amely azt mutatja, hogy a pályába helyezett pont előtt milyen távolságban jelenjen meg a **Marker**. A **Marker** az adott [szelvény számnál](#) tűnik el.

## Track.PointOfInterest

**Track.PointOfInterest** *VágányIndex; X; Y; ElforgatásY; ElforgatásX; ElforgatásZ; Név*

**Track.POI** *VágányIndex; X; Y; ElforgatásY; ElforgatásX; ElforgatásZ; Név*

A parancs egy kamerát helyez el a pálya mentén. A kamerák között a NUM 1 és NUM 7 gombokkal lehet lépkedni.

**VágányIndex:** Annak a vágálynak az indexszáma, amelyhez viszonyítva a kamera elhelyezésre kerül.

**X:** **Vízszintes** távolság a **VágányIndexben** meghatározott vágánytól.

**Y:** **Függőleges** távolság a **VágányIndexben** meghatározott vágánytól.

**ElforgatásY:** A kamera elforgatásának szöge **fokban**, az **Y tengely** körül (azaz balra/jobbra). Másképpen: az elforgatás szöge az **XZ-síkot felülről nézve**, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.

**ElforgatásX:** A kamera elforgatásának szöge **fokban**, az **X tengely** körül (azaz fel/le). Másképpen: az elforgatás szöge az **YZ-síkot balról nézve**, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.

**ElforgatásZ:** A kamera elforgatásának szöge **fokban**, a **Z tengely** körül (azaz az objektum hossz tengelyén nézve balra/jobbra). Másképpen: az elforgatás szöge az **XY-síkot hátulról nézve**, az óra járásával megegyező irányban. Az alapértelmezett érték **0**.

**Név:** A kamera neve

## Track.PreTrain

Nincs leírás.

## Track.Announce

**Track.Announce** *Fájlnev*

A parancs lejátszik egy hangfájlt.

**Fájlnev:** A lejátszandó hangfájl neve, a **Sound** mappához viszonyított **relatív útvonalon** kell megadni.

## Track.Doppler

Nincs leírás.

## Track.Buffer

**Track.Buffer**

Ez a parancs egy olyan pontot helyez el a pályán, amelyen a járművel nem lehet áthaladni. Ez a pont megállítja a járművet mindkét irányból. Ütközőbakként alkalmazható a pálya elején és végén, azonban ez a parancs vizuálisan nem helyez el objektumot. Amennyiben objektum szükséges, akkor a [Track.Freeobj](#) parancssal helyezhető el. A parancsnek nincsenek paraméterei.



## Összefoglaló

### Tudnivalók:

A **narancssárga** színnel kiemelt parancsok kerülnek feldolgozásra legelőször, a többi parancs feldolgozása csak ezt követően történik. A **zöld** színnel kiemelt paraméterek az [Options.UnitOfLength](#) parancs beállításaitól, a **kék** színnel kiemelt paraméterek az [Options.UnitOfSpeed](#) parancs beállításától függenek.

A nem kötelező (elhagyható) utótagok szürke színnel, a **kötelező** utótagok **félkövér** betűvel vannak jelezve.

### Szelvényszám

Szelvényszám

Rész<sub>1</sub> : Rész<sub>2</sub> : ... : Rész<sub>n</sub>

### Helyettesítők, véletlenszám generálók

\$Chr(Ascii)

\$Rnd(Minimum; Maximum)

\$Sub(Index) = Érték

\$Sub(Index)

### „Options” szakasz

**Options.UnitOfLength** FaktorMéter<sub>1</sub>; FaktorMéter<sub>2</sub>; ...; FaktorMéter<sub>n</sub>

**Options.UnitOfSpeed** FaktorKMH

**Options.BlockLength** Hossz

**Options.ObjectVisibility** { 0 = hagyományos | 1 = szelvényalapú }

**Options.SectionBehavior** { 0 = index-alapú | 1 = érték-alapú }

**Options.FogBehavior** { 0 = blokk-alapú | 1 = lineáris }

### „Route” szakasz

**Route.Comment** Szöveg

**Route.Image** Fájlnév

**Route.Timetable** Szöveg

**Route.Change** { -1 = üzemi fék (ATS be) | 0 = vészfék (ATS ki) | 1 = vészfék (ATS be) }

**Route.Gauge** ÉrtékMilliméter

**Route.Signal(Index).Set** Sebesség

**Route.RunInterval** ÉrtékMásodperc

**Route.AccelerationDueToGravity** Érték

**Route.Elevation** Magasság

**Route.Temperature** ÉrtékCelsius

**Route.Pressure** ÉrtékKPa

**Route.AmbientLight** Vörös; Zöld; Kék

**Route.Directionallight** Vörös; Zöld; Kék

**Route.LightDirection** Theta; Phi

### „Train” szakasz

**Train.Folder** Mappanév

**Train.Run(SínTípusIndex).Set** RunHangIndex

**Train.Flange(SínTípusIndex).Set** FlangeHangIndex

**Train.Timetable(MenetrendIndex).Day** Fájlnév

**Train.Timetable(MenetrendIndex).Night** Fájlnév

**Train.Gauge** ÉrtékMilliméter

**Train.Interval** ÉrtékMásodperc

### „Structure” szakasz

**Structure.Rail(SínTípusIndex).Load** Fájlnév

**Structure.Beacon(BeaconIndex).Load** Fájlnév

**Structure.Pole(VágányokSzáma; OszlopIndex).Load** Fájlnév

**Structure.Ground(TalajIndex).Load** Fájlnév

**Structure.Wall(FallIndex).Load** Fájlnév

**Structure.WallR(FallIndex).Load** Fájlnév

**Structure.DikeL**(TöltésIndex).Load FájlNév  
**Structure.DikeR**(TöltésIndex).Load FájlNév  
**Structure.FormL**(PeronIndex).Load FájlNév  
**Structure.FormR**(PeronIndex).Load FájlNév  
**Structure.FormCL**(PeronIndex).Load FájlNév  
**Structure.FormCR**(PeronIndex).Load FájlNév  
**Structure.RoofL**(TetőIndex).Load FájlNév  
**Structure.RoofR**(TetőIndex).Load FájlNév  
**Structure.RoofCL**(TetőIndex).Load FájlNév  
**Structure.RoofCR**(TetőIndex).Load FájlNév  
**Structure.CrackL**(KitöltésIndex).Load FájlNév  
**Structure.CrackR**(KitöltésIndex).Load FájlNév  
**Structure.FreeObj**(ObjektumIndex).Load FájlNév

### „Texture” szakasz

**Texture.Background**(HáttérképTextúraIndex).Load FájlNév  
**Texture.Background**(HáttérképTextúraIndex).X Darabszám  
**Texture.Background**(HáttérképTextúraIndex).Aspect { 0 = fix magasság | 1 = eredeti képméretarány }

### „Cycle” szakasz

**Cycle.Ground**(TalajIndex).Params TalajIndex<sub>0</sub>; TalajIndex<sub>1</sub>; ...; TalajIndex<sub>n</sub>

### „Signal” szakasz

**Signal**(FényjelzőIndex).Load FényjelzőFájlKiterjesztésNélkül; IzzóFájlKiterjesztésNélkül  
**Signal**(FényjelzőIndex).Load AnimáltObjektumFájl

### „Track” szakasz

Vágányok:

**Track.RailStart** VágányIndex; X; Y; SínTípusIndex  
**Track.Rail** VágányIndex; X; Y; SínTípusIndex  
**Track.RailType** VágányIndex; SínTípusIndex  
**Track.RailEnd** VágányIndex; X; Y  
**Track.Accuracy** Érték  
**Track.Adhesion** Érték

Vonalvezetés:

**Track.Curve** Ívsugár; TúlemelésMilliméter  
**Track.Pitch** ÉrtékEzrelék  
**Track.Turn** Érték  
**Track.Height** Magasság

Objektumok:

**Track.FreeObj** VágányIndex; ObjektumIndex; X; Y; ElforgatásY; ElforgatásX; ElforgatásZ; Térköz  
**Track.Wall** VágányIndex; Oldal; FallIndex  
**Track.WallEnd** VágányIndex  
**Track.Dike** VágányIndex; Oldal; TöltésIndex  
**Track.DikeEnd** VágányIndex  
**Track.Pole** VágányIndex; VágányokSzáma; Oldal; Intervallum; OszlopIndex  
**Track.PoleEnd** VágányIndex  
**Track.Crack** VágányIndex<sub>1</sub>; VágányIndex<sub>2</sub>; KitöltésIndex  
**Track.Ground** CycleIndex

*Állomások:*

**Track.Sta** *ÁllomásNév; Érkezésildő; Indulásildő; ÁthaladásFigyelmeztetés; AjtóOldal; Kijáratijelző; BiztonságiRendszer; ÉrkezésHang; Tartózkodásildő; Utasmennyiség; IndításjelzőHang; MenetrendIndex*

**Track.Station** *ÁllomásNév; Érkezésildő; Indulásildő; Kijáratijelző; BiztonságiRendszer; IndításjelzőHang*

**Track.Stop** *Oldal; HatárHátra; HatárElőre; KocsikSzama*

**Track.Form** *VágányIndex<sub>1</sub>; VágányIndex<sub>2</sub>; TetőIndex; PeronIndex*

*Jelzések:*

**Track.Limit** *Sebesség; Oldal; Irány*

**Track.Section** *Térköz<sub>0</sub>; Térköz<sub>1</sub>; ...; Térköz<sub>n</sub>*

**Track.SigF** *Típus; Térköz; X; Y; ElforgatásY; ElforgatásX; ElforgatásZ*

**Track.Signal** *Típus; Szöveg; X; Y; ElforgatásY; ElforgatásX; ElforgatásZ*

**Track.Relay** *X; Y*

*Biztonsági rendszerek:*

**Track.Beacon** *Type; BeaconStructureIndex; Section; Data*

**Track.Transponder** *Type; Signals; SwitchSystems*

**Track.AtsSn**

**Track.AtsP**

**Track.Pattern** *Type; Speed*

**Track.PLimit** *Sebesség*

*Egyéb:*

**Track.Back** *HáttérképTextúraIndex*

**Track.Fog** *TávolságKezdet; TávolságVége; Vörös; Zöld; Kék*

**Track.Brightness** *Érték*

**Track.Marker** *FájlNév; Távolság*

**Track.PointOfInterest** *VágányIndex; X; Y; ElforgatásY; ElforgatásX; ElforgatásZ; Név*

**Track.PreTrain** *Idő*

**Track.Announce** *Fájl; Sebesség*

**Track.Doppler** *FájlNév; X; Y*

**Track.Buffer**

## Gyakoribb hibaüzenetek

### Route fájl hibák

---

#### *Típus Fájlnév not found*

A **Fájlnév**-ben leírt adott **Típusú** fájl a megadott útvonalon nem található. Ellenőrizzük, hogy a fájlnev jó-e, és az elérési útvonal megfelelő-e. (példa a hibaüzenetre: Object file **Fájlnév** not found)

---

#### **Index is required to reference a non-existing rail in Track.Railstart**

A **Track.Railstart** parancsban használt indexszám olyan lehet, amely éppen nem használatos. Például ha egymást követő blokkban két Railstart parancsot használunk **ugyanazzal** a vágány indexszámmal. A régi BVE nem jelezte ezt a hibát, a második Railstart parancsot Rail parancsnak vette.

---

#### **IdxStType references an object not loaded in Track.parancsnév**

A **Track.parancsnév**ben használt indexszám nincs definiálva a **Structure.parancsnév** szakaszban.

---

#### **The number of aspects is not supported in Track.Signal**

A fényjelzőhöz használt Típus szám nem jó a **Track.Signal** parancsban. A korábbi BVE-ben használható volt **0** és **1** érték is. Az openBVE-ben -2, 2, 3, -4, 4, -5, 5, 6 értékek lehetnek, 0 és 1 nem.

---

#### **A stop without a station is invalid in Track.Stop**

**Track.Stop** parancs **Track.Sta** parancs nélkül. A route fájl elején a **Track.Stop** parancsot hamarabb alkalmaztuk, mint **Track.Sta** parancsot.

---

### Objektum hibák

---

#### **The Command *parancs* is not supported; The attribute *parancs* is not supported**

A **parancs** helyén leírt szövegrészt parancsként értelmezi a program, de ilyen parancs nem létezik. Ellenőrizzük a karaktereket, hogy megfelelő-e a parancsnév.

---

#### **The direction indicated by X, Y and Z is expected to be non-zero in *parancs***

Az adott **Parancsban** az X, Y, Z értékek helyének **mindegyikén** 0-t írtunk, de **nem lehet mindegyik** 0. Valahol 1-nek kell lennie. Ha nem akarjuk a parancssal módosítani az adott területet, akkor töröljük a parancsot.

---

#### **At most *x* arguments are expected in *parancs***

Az adott **parancsban** legfeljebb az **x**-ben kiadott mennyiségű paraméter alkalmazható. Ellenőrizzük, hogy nem-e több paraméter van megadva, és ennek megfelelően javítsuk ki az adott sort.

---

#### **0 arguments are excepted in [Meshbuilder]**

A meshbuilder szöveg után írtunk valamit, csak nem elé tettünk pontosvesszőt, hanem a szöveg mögé. Ezért ezt a szöveget is parancsként értelmezi az openBVE. Írjuk át megfelelőre a sort.



## Egyéb fogalmak

### úrszelvény

Az a pályaszerkezet feletti szabványosított keresztmetszeti terület, amelyet a vasúti forgalom biztonságos lebonyolításához szabadon kell tartani.

### kitérő

A **kitérő (váltó)** a vágányba épített olyan szerkezet, amely lehetővé teszi a vasúti járművek egyik vágányról a másikra történő folyamatos áthaladását.

Normál [nyomtávolságú](#) pályák esetén a szokásos, közönséges váltósugár **R = 200 méter** körüli érték.

A jármű mozgása során létrejövő centrifugális oldalgyorsulás értéke szabja meg a váltók eltérítőágában alkalmazható sebességet. Ez az oldalgyorsulás itt a [túlemelés](#) hiánya miatt teljes mértékben érvényre jut. A megengedett oldalgyorsulási érték azonos a folyópálya csökkentett túlemelése esetén megengedett szabad oldalgyorsulással:

$$P_e = 0,6 \text{ m/s}^2$$

A váltóban **R** sugár mellett megengedhető sebesség (km/h) értéke:

$$V_{\max} = \sqrt{0,6 \cdot 13R} = 2,8\sqrt{R}$$

A képlet szerint egy **200 méteres sugarú kitérőben** a legnagyobb sebesség  $v_{\max} = 2,8 \times 14,2 \approx 40 \text{ km/h}$  lehet.

### túlemelés

**Meghatározása:** az ívben fellépő oldalgyorsulás csökkentésére a külső sínszál kiemelése. Megoldható úgy is, hogy a külső sínszál emelése mellett a belső sínszálát egyidejűleg süllyesztik. Kifuttatása az átmeneti ívben történik.

Az egyenes pályaszakaszokon a két sínszál azonos magasságban fekszik. A pályán gördülő járművek súlya egyformán terheli mindkét sínszálát. Az íves pályaszakaszon haladó járműre a súlyerőn kívül a vízszintes irányú centrifugális erő is hat. A két erő eredője nem lesz merőleges a pályasíkra. Ennek káros következményei a következők:

- a külső sínszál belső vezetési felületén, a megnövekedett terhelés következtében jelentkező erős oldalkopás,
- a sínlekötések gyorsabb elhasználódása,
- a vontatási ellenállás megnövekedése,
- kényelmetlen utazás, áru rongálódás.

Ezeket a káros erőhatásokat lehet csökkenteni ill. kiküszöbölni a külső sínszál megemelésével, az úgynevezett **túlemeléssel**.

A túlemelés általi jármű megdöntés által a jármű megdöntés szöge azonos a nehézségi gyorsulás és az ívben fellépő (a sebességtől és az ívsugártól függő) centrifugális erő eredőjének a függőlegessel bezárt szögével. Ezen eredő vektora merőleges kell hogy legyen a kerekek felfekvő pontjának összekötött vonalával. Ekkor van az oldalirányú erő a túlemeléssel kiegyenlítve.

**Szabványos túlemelés mértékének (m) meghatározása,** ahol

$x$  = a [nyomtávolság](#),  $V$  = a jármű sebessége,  $R$  = a körív sugara:

$$m = \frac{V^2 \cdot (x + 65)}{R \cdot 9,81 \cdot 3,6^2} * 0,85$$

**Megjegyzés:** A valóságban nem kell az alapérték 0,85-szeresét kiszámolni. A BVE 2 és BVE 4 a valós túlemelés értékének kb. 0,85-szeresénél jelenít meg valós megdőlést. A képlet arra az esetre vonatkozik, amikor sebesség és az ív függvényében az oldalgyorsulást a túlemeléssel teljes mértékben kiegyenlítik (elméleti túlemelés).

## Példák

### Példa az Options.ObjectVisibility módjainak különbségeiről

25 méteres blokk esetén (**Options.BlockLength 25**) a 1234. szelvényben elhelyezett **15 méter hosszú** objektum **0 esetén** 1250-nél válik láthatatlanná (az objektum teljesen látható marad).

**1 esetén** 1249-nél válik láthatatlanná (az objektum teljesen látható marad).

10 méteres blokk esetén (**Options.BlockLength 10**) a 1234. szelvényben elhelyezett **15 méter hosszú** objektum **0 esetén** 1240-nél válik láthatatlanná (az objektumból látható **9 méternyi** rész hirtelen eltűnik).

**1 esetén** 1249-nél válik láthatatlanná (az objektum teljesen látható marad).

1 méteres blokk esetén (**Options.BlockLength 1**) a 1234. szelvényben elhelyezett **15 méter hosszú** objektum **0 esetén** 1235-nél válik láthatatlanná (az objektumból látható **14 méternyi** rész hirtelen eltűnik).

**1 esetén** 1249-nél válik láthatatlanná (az objektum teljesen látható marad).

### Példa a Track.Brightness parancs alkalmazására

A következő példában 75 méter alatt világosból sötét lesz a környezet:

```
1000, Track.Brightness 255
1075, Track.Brightness 0
```

Ezután ha később a sötétebből világos környezeti hatást szeretnénk elérni, akkor az előző befejező értékkel kezdjük, és adjuk meg az új fényerő értéket:

```
1275, Track.Brightness 0
1300, Track.Brightness 255
```

Így a pályában a környezet 1000-1075 között sötétedni fog, 1075-1275 között változatlanul sötét marad, majd 1275-1300 között fokozatosan világosodni fog.

### Példa a Track.Stop parancs alkalmazására

A következő példában mindegyik járművel a 1120. szelvénynél kell megállni. Hátrafelé és Előrefelé is 3-3 méter távolságon belül (azaz 1117-1123 szelvények között):

```
1000, Track.Sta állomás1;...
1120, Track.Stop 0;3;3
```

A következő példában a **3 kocsis** járművel a 1060. szelvénynél, **Hátrafelé 3 méter, Előrefelé 2 méter távolságon belül** (azaz 1057-1062. szelvények között) kell megállni.

A **6 kocsis** járművel a 1120. szelvénynél kell megállni **Hátrafelé és Előrefelé is 2.5-2.5 méter távolságon belül** (azaz 1118-1122. szelvények között) kell megállni:

```
1000, Track.Sta állomás1;...
1060, Track.Stop 0;3;2;3
1120, Track.Stop 0;2.5;2.5;6
```

### Példa a Route.Signal, Track.SigF és Track.Section használatára

Példa *index-alapú térközműködésre* a *Route.Signal*, *Track.SigF* és *Track.Section* parancsok alkalmazásával:

**Jelzési képek, illetve az azokhoz rendelt sebességértékek meghatározása**

#### 1. „Route” szakaszba tartozó parancsok beírása

A fájl elején a **Route szakaszban** az indexszámokhoz sebességértékeket rendelünk. A jelzési képeket saját magunk határozhatjuk meg, **de a 0 indexszám mindig a vöröset jelenti.**

(a pontosvessző után egy példa a jelzési képekre, és a sebességértékeikre)

```
Route.Signal(0) 0 ;egy vörös fény amely nem haladható meg
Route.Signal(1) 40 ;két sárga fény amely 40 km/h-val haladható meg
Route.Signal(2) 50 ;egy sárga fény amely 50 km/h-val haladható meg
Route.Signal(3) 60 ;zöld-sárga fény amely 60 km/h-val haladható meg
Route.Signal(4) 70 ;egy zöld fény amely 70 km/h-val haladható meg
```

#### 2. Jelzési képek textúráinak elkészítése

A fényjelzőtípusokhoz és azok jelzési képeihez textúrákat kell készíteni. Az izzó fényjelzőtípusban más textúrát célszerű használni, amelyben a világító optika fényköre nagyobb átmérővel rendelkezik. Így ez messzebről is látszódnia

fog a játékban, mert olyan hatást kelt. Az adott jelzési kép textúrafájl neveinek **utolsó karakternek meg kell egyeznie a Route.Signal parancsokban használt jelzési képek indexszámával.**

(zárójelben a magyarázat)

Signal_2optikas_0.bmp	(2 optikás jelző, amelyen <b>egy vörös</b> fény világít)
Signal_2optikas_2.bmp	(2 optikás jelző, amelyen <b>egy sárga</b> fény világít)
Signal_2optikas_4.bmp	(2 optikás jelző, amelyen <b>egy zöld</b> fény világít)
Signal_3optikas_0.bmp	(3 optikás jelző, amelyen <b>egy vörös</b> fény világít)
Signal_3optikas_1.bmp	(3 optikás jelző, amelyen <b>két sárga</b> fény világít)
Signal_3optikas_2.bmp	(3 optikás jelző, amelyen <b>egy sárga</b> fény világít)
Signal_3optikas_3.bmp	(3 optikás jelző, amelyen <b>zöld-sárga</b> fény világít)
Signal_3optikas_4.bmp	(3 optikás jelző, amelyen <b>egy zöld</b> fény világít)
Signal_4optikas_0.bmp	(4 optikás jelző, amelyen <b>egy vörös</b> fény világít)
Signal_4optikas_1.bmp	(4 optikás jelző, amelyen <b>két sárga</b> fény világít)
Signal_4optikas_2.bmp	(4 optikás jelző, amelyen <b>egy sárga</b> fény világít)
Signal_4optikas_3.bmp	(4 optikás jelző, amelyen <b>zöld-sárga</b> fény világít)
Signal_4optikas_4.bmp	(4 optikás jelző, amelyen <b>egy zöld</b> fény világít)

### 3. Jelzési képek objektumainak elkészítése

Készítsünk a **fényjelzőtípusokhoz** objektumkat, külön normál fényjelzőt és izzó fényjelzőt.

A példa szerint 3 fényjelzőtípusunk van (2 optikás, 3 optikás és 4 optikás).

Az objektumban mindegy, hogy melyik textúrát adjuk meg, mert a program a későbbiekben ezt változtatja.

A fájlokat úgy kell elnevezni, hogy **az előbb használt textúra nevek utolsó karakterét figyelmen kívül kell hagyni:**

```
Signal_2optikas_.b3d
Signal_3optikas_.b3d
Signal_4optikas_.b3d
Izzo_2optikas_.b3d
Izzo_3optikas_.b3d
Izzo_4optikas_.b3d
```

**A leírás nem tartalmazza az objektumfájlok elkészítésének menetét!**

### 4. A jelzőtípusok definiálása a Structure szakaszban

A Structure **szakasz végére(!)** az indexszámokhoz fényjelzőtípusokat és izzó fényjelzőtípusokat rendelünk.

**A „Signal” szó előtt pontot nem kell tenni és a fájl kiterjesztését nem kell megadni!**

```
Signal(2) en_palyam\signals\Signal_2optikas ; Izzo_2optikas_
Signal(3) en_palyam\signals\Signal_3optikas ; Izzo_3optikas_
Signal(4) en_palyam\signals\Signal_4optikas ; Izzo_4optikas_
```

### 5. A Track.SigF és Track.Sextion parancsok alkalmazása a „Track” szakaszban

A Track szakaszban a SigF és a Section parancs használata együtt:

```
1000, SigF 3;1;1.9;2.2, .Section 0;0;2;3;4
```

**SigF parancs értelmezése:**

A Structure részben definiált **3-as indexszámú fényjelzőtípust** (3 optikás) helyezük el, a vágánytól 1,9 méterre jobbra, és 2,2 méterre felfelé. **A SigF parancs 2. paraméterének működése ismeretlen.**

**Section parancs értelmezése:**

- Amennyiben a **fényjelző mögött lévő térköz** foglalt, akkor a **3-as indexszámú fényjelzőtípus** azon textúrája fog megjelenni, **amelyen egy vörös fény világít** (Signal\_3optikas\_0.bmp). Illetve a **Route.Signal(0)** parancsához megadott értékkel haladhatjuk meg a fényjelzőt. A 0 mindig vöröset jelent. Ha a fényjelzőt meghaladjuk, akkor figyelmeztető üzenet jelenik meg a játék közben.
- Amennyiben a **fényjelző mögött lévő 2. térköz** foglalt, akkor a **3-as indexszámú fényjelzőtípus** azon textúrája fog megjelenni, **amelyen egy vörös fény világít** (Signal\_3optikas\_0.bmp). Valamint a **Route.Signal(0)** parancsához megadott értékkel haladhatjuk meg a fényjelzőt. A példa szerint ebben az esetben még ekkor sem haladhatjuk meg a fényjelzőt. Ha a fényjelzőt meghaladjuk, akkor figyelmeztető üzenet jelenik meg a játék közben.
- Amennyiben a **fényjelző mögött lévő 3. térköz** foglalt, akkor a **3-as indexszámú fényjelzőtípus** azon textúrája fog megjelenni, **amelyen egy sárga fény világít** (Signal\_3optikas\_2.bmp). Valamint a **Route.Signal(2)** parancsához megadott értékkel haladhatjuk meg a fényjelzőt. Ha a fényjelzőt az ehhez az indexszámhoz rendelt magasabb sebességgel haladjuk meg, akkor figyelmeztető üzenet jelenik meg a játék közben.
- Amennyiben a **fényjelző mögött lévő 4. térköz** foglalt, akkor a **3-as indexszámú fényjelzőtípus** azon textúrája fog megjelenni, **amelyen zöld-sárga fény világít** (Signal\_3optikas\_3.bmp). Valamint a **Route.Signal(3)**

parancsához megadott értékkel haladhatjuk meg a fényjelzőt. Ha a fényjelzőt az ehhez az indexszámhoz rendelt magasabb sebességgel haladjuk meg, akkor figyelmeztető üzenet jelenik meg a játék közben.

- Amennyiben a **fényjelző mögött lévő 5. térköz** foglalt, akkor a **3-as indexszámú fényjelzőtípus** azon textúrája fog megjelenni, **amelyen egy zöld fény világít** (Signal\_3optikas\_4.bmp). Valamint a **Route.Signal(4)** parancsához megadott értékkel haladhatjuk meg a fényjelzőt. Ha a fényjelzőt az ehhez az indexszámhoz rendelt magasabb sebességgel haladjuk meg, akkor figyelmeztető üzenet jelenik meg a játék közben.



<http://www.bveklub.hu/>

<http://www.bvemetro.hu/>

© 2009