



Itt a vége, fuss el véle...

Gimpről szóló cikksorozatunk a befejező részéhez érkezett, s reméljük, olvasóink kellő bepillantást nyerhettek a webes felületre szánt grafikák elkészítésének rejtelmeibe.

E zárórészben áttekintjük az eddigi témákat, összefoglaljuk a tulajdonságokat és befejezzük a maszkolás témakörét, majd rövid bevezetést nyújtok a belső függvények készítésébe.

A kép háttérének megváltoztatása

Az előző cikkben szereplő elméleti bevezető után most a maszkolás gyakorlati részével foglalkozunk. Alkotásaink létrehozása során gyakran szükségünk lehet a háttér módosítására, ezért megnézzük, miként is változtathatjuk meg. Vessünk egy pillantást az **1. képre!**

Az elkészült kép jól sikerültnek nevez-



1. kép A kiindulási kép



2. kép A kép háttérének módosítása

hető, de a háttér bal szélén nyomasztóan hat a szikla, így hát lecseréljük. Első lépésként takarjuk ki a cserélendő területet, amelyet egy új csatornában ajánlott elkészíteni – ha így cselekszünk, később akár kijelölésként, akár maszkként is gyorsan alkalmazhatjuk. A maszkban mindenki kedve szerint, valamint megszokott eszközei segítségével készít-

heti el a kitakarást, ennek menetét sorozatunk előző részében mutattam be. A cikkben szereplő kép szerkesztését időközben befejeztem, a munka végeredménye a **2. képen** látható. Háttérként szándékosan választottam ezt a kissé meglepő képet, ezáltal ugyanis jobban elválik a korábbi és a mostani háttér. A „mű” még egyáltalán nincsen készen, például a szörfös vitorláján lévő ablakon át látszik a korábbi háttér. A hibák felsorolását folytathatnám, de gondolom, az olvasók még nálam is jobban látják ezeket. Szerencsére a Gimp kínálta maszkolásnak köszönhetően mindent kijavíthatunk. Hozunk létre egy újabb maszkot, amely a szörfös vitorlájának ablakát tartalmazza, ezen tudjuk majd az átlátszóságot a kívánt mértékűre igazítani. A feladat megoldását olvasóinkra bízom, a munka elvégzéséhez szükséges fájlokat mind a CD-mellékleten, mind az Interneten megtalálhatjátok. Az elkészült képeket a Világhálón fogjuk „kiállítani”.

A beépített függvények bővítése

A sorozatunk első részében számba vett szolgáltatások közül a Gimp egyik legelőnyösebb és legnagyobb témakörével eddig még nem foglalkoztunk. Többek kérésére most a Script-Fu függvények készítését tekintjük át röviden. E témakört több szempontból is sorozatunk végére hagytam:

1. Nem a programozással kezdtem, hogy a nem programozói beállítottságú olvasók se riadjanak vissza a Gimp használatától.
2. A Gimp programozásához jól kell ismerni a program működését.
3. A sorozat elsődleges célja nem a programozás elsajátítására, hanem a Linux grafikai lehetőségeinek kihasználására irányult.

Tehát figyelem, kevésbé „programozókedvű” olvasók is nyugodtan tovább olvashatják írásunkat.

Mi a Script-Fu és hogyan lehet használni?

A Script-Fu elnevezés nem japán étel vagy ital, hanem a Gimp Lisp-alapú,

beépített programozói felületét takarja. A felépítés, valamint a programozási nyelvek fejlődésének részletes ismertetésétől terjedelmi okok miatt most tekintsünk el. A Script-Fu használatára két alkalommal nyílik lehetőségünk:

1. A Gimp belső parancsainak ismeretében kis programunkat szövegszerkesztő segítségével készítjük el (gyakorlott programozók a cat segítségével is írhatják).
2. A konzol (**3. kép**) és a DB-böngésző (**4. kép**) segítségével összeállítjuk a programot, majd a kódot kedvenc szövegszerkesztőnkkel készítjük el.

Krómfelirat készítése a Script-Fu konzolon

A következő részben a feliratot szerkesztjük meg a Script-Fu konzolon. A rajzok elkészítése során már említettem, hogy alkotáskor az összetett feladatokat érdemes kisebb, a programban végrehajtható lépésekre bontani. Ez az állítás a függvények estében még fokozottabban érvényes. A konzol (**3. kép**) alsó sorába írhatjuk be a parancsokat. Ha valamelyik parancs felépítésével vagy működésével nem vagyunk teljesen tisztában, a Tallózás (Browse) gombra kattintva megnézhetjük a parancshoz tartozó leírást (**4. kép**).

Gépeljük be a következő parancsot: (gimp-image-new 200 200 RGB)



3. kép A Script-Fu konzol

ekkor a konzolon az alábbi üzenetet látjuk:

```
=> (gimp-image-new 200 200 RGB)
(0)
```

Az első sorban láthatjuk a végrehajtott utasítást, a másodikban pedig az áltál visszaadott értéket. Mit is csinált ez az utasítás? Létrehoztunk egy 200×200 méretű RGB-képet. De mi a (0)? A DB-böngészőben azt olvashatjuk, hogy ez a kép hivatkozási száma. Remek, sikerült elkészítenünk képet, viszont nem látjuk! Megszoktuk, hogy kép a létrehozása után azonnal látható lesz. Itt azonban a belső szűrők, kiegészítők miatt a képet



4. kép A beépített modulok



5. kép A Gimpben található krómfelületű függvény eredménye

láthatóvá kell tennünk. Az elején nehézséget okozhat nekünk, hogy azok a szolgáltatások, amelyeket a programban megszoktunk, itt külön működnek, tehát ahhoz, hogy a képet láthatóvá tessük, egy réteget kell hozzáadnunk:

```
(gimp-layer-new 0 200 200
↳RGB-IMAGE "reteg"
↳100 NORMAL-MODE)
```

```
=> gimp-layer-new 0 200 200
↳RGB-IMAGE "reteg"
↳100 NORMAL-MODE
(2)
```

Sikerült! A parancs által visszaadott érték (2) a réteg hivatkozási száma.

Tegyük láthatóvá a réteget:

```
(gimp-image-add-layer 0 2 0)
=> gimp-image-add-layer 0 2 0
(1)
```

Majd a képet is:

```
(gimp-display-new 0)
=> (gimp-display-new 0)
(1)
```

Senki ne ijedjen meg, nem rontott el semmit – a réteg csak a memóriában található „szeméttel” van feltöltve, ezért meg kell tisztítanunk:

```
(gimp-drawable-fill
↳2 GB-IMAGE-FILL)
```

```
=> (gimp-drawable-fill
↳2 BG-IMAGE-FILL)
```

(1) Ennél a szolgáltatásnál ügyeljünk rá, hogy a réteg rajzolható legyen. Ha a fenti utasításokat a kódba szeretnénk írni, az alábbi módon tehetjük meg:

```
(define (my-make-new-image)
(let* ((image (car
↳(gimp-image-new 200 200 RGB)))
(layer (car
↳(gimp-layer-new image
↳200 200 RGB-IMAGE "reteg"
↳100 NORMAL-MODE))))
(gimp-drawable-fill
↳layer BG-IMAGE-FILL)
(gimp-image-add-layer
↳image layer 0)
(gimp-display-new
↳image)
image))
```

Íme, első Script-Fu függvényünk – még nincs kész, de máris nagyon ígéretes.

Írjuk bele a következő szöveget:

```
(gimp-text-fontname 0 3 0 0
↳"Linux" 0 1 25 PIXELS
↳"-abisource-courier-regular-
↳r-normal--*-*-75-75-p-90
↳-iso8859-1")
=> (gimp-text-fontname 0 3 0 0
↳"Linux" 0 1 25 PIXELS
↳"-abisource-courier-regular
↳r-normal--*-*-75-75-p-90
↳-iso8859-1")
(3)
```

Már olvashatjuk is. (Figyelem, a fenti betűkészlet nem biztos, hogy más gépen is működik!) Készítsünk másolatot a rétegről, mivel azonban a DB-böngészőben nincs ilyen eljárás, most létre kell hoznunk egyet:

```
(define (my-duplicate-layer
↳image layer)
(let* ((dup-layer
↳(car (gimp-layer-copy layer
↳1))))
(gimp-image
↳-add-layer image dup-layer 0)
↳dup-layer))
```

Miután az eljárást beírtuk a konzolba (egy sorba), hajtsuk is végre:

```
=> (my-duplicate-layer 0 3)
(8)
```

A következő lépésben méretezzük át a réteget:

```
(gimp-layer-resize 8 200 100 5 5)
Ezután az alfacsatorna szerint jelöljük ki, majd szövegünkhöz adjuk hozzá a vastag keretet is:
```

```
(gimp-selection-layer-alpha 8)
(gimp-selection-grow 0 5)*
A kép kiszínezése előtt rétegünket
```

```
(gimp-layer-set-preserve
↳-trans 8 0)
=> (gimp-blend 8 CUSTOM
↳NORMAL LINEAR 100 0 REPEAT
↳-NONE FALSE 0 0 0 0 30 50)
(1)
```

Parancsfájlunk számára minden apró kis építőköckökre rendelkezésre áll, tehát nincs más hátra, mint leírunk. Miután parancsfájlunk szövegszerkesztőbe vitelét sikeresen befejeztük, felmerül a kérdés: miként vehetjük rá a Gimpet, hogy használja is? A saját Script-Fu függvényeket a `~/gimp-1.2/scripts/` könyvtárban tároljuk. A Gimp ezeket minden indulás után itt keresi, és ha a formátumuk megfelelő, azonnal használatba is vehetők. Mi szükséges ahhoz, hogy használható kódot kapjunk? A DB-böngészőben használt utasításokat adott formai követelményeknek megfelelően kell leírunk. Kódunk végére az alábbi szöveget másoljuk be:

```
(script-fu-register
↳"krómfelirat keszítése"
↳_ "<Toolbox>/Xtns/Script
↳-Fu/Egyeb/Krómfelirat..."
↳"Felirat keszítése"
↳színatmenettel"
↳"Gabor Suveg"
↳"Gabor Suveg"
↳"Sept 2001"
↳" "
↳SF-STRING "Text"
↳"Linux"
↳SF-FONT "Font" " *-
↳bitwise alpha--*-r--*-*-
↳*-*-*-*-*"
↳SF-VALUE "Font Size"
↳"50" )
```

Az elkészült program

A fenti leírás csak betekintést kívánt nyújtani a Gimp programozási lehetőségeibe. A fenti programon még rengeteg dolgot kell javítani, illetve csiszolni. A továbbfejlesztett, kijavított szűrőt a *Kit./Script-Fu/Logók/Króm...* alatt találhatjuk meg (lásd az 5. képen).

Ezennel Gimpvel foglalkozó sorozatunk a végéhez érkezett. További leírások és ismertetőik elérhetők lesznek a

☞ <http://www.gimp.hu> oldalon, és a későbbiekben is szívesen válaszolok minden felmerülő kérdésre.

Köszönöm a figyelmet!



Süveg Gábor
(gsuveg@sgsystem.com)
Régóta használ Linuxot és BSD-t. Hobbija a bűvárkodás, a vitorlázás és a számítógépes grafika.