PCLinuxOS Magazine – 2014. július

Írta: Peter Kelly (critter)

Megjegyzések

Az előző cikkben azt állítottam, hogy a gtkdialog kódjának XML részében rendesen nem engedélyezett megjegyzések készítése, de egy apró trükkel megoldható. A leghatásosabb módjának a Puppy Linux fórumon zigbert kiváló sorozatában felvázolt eljárást találtam.

Először, olyan karaktersort kell választani a megjegyzés indítójának, ami a kódban normálisan nem jelenik meg. A bash-héj a kettős keresztet használja a megjegyzéshez, de a gtkdialog a színek hexa kódját gyakran használja, ami kettős kereszttel kezdődik. Ennek kikerülésére zigbert dupla kettős kereszt "##" bevezetését javasolja, én magam is követem ezt a gyakorlatot, de te azt választhatsz, amit akarsz addig, amíg az eljárás számára egyértelmű.

A színek gyakran hexadecimális formában adottak, és kettős keresztet "#" követően jelennek meg. A hexadecimális rendszer tizenhatig számol úgy, ahogy rendesen számolunk tízig (decimális). A tizenhat érték megjelenítésére a 0-9 számjegyek és A-F (vagy a-f) betűk szolgálnak.

A modern digitális számítógépek, működési módjuk következtében sokkal hatékonyabbak, amikor binárisan (kettes), oktálisan (nyolcas), vagy hexadecimális számokat kapnak.

Egy színt a piros, a zöld és a kék összetevők, vagy csatornák értékével lehet meghatározni. Két hexadecimális szám alkalmazása minden színösszetevőre 256 lehetőséget ad, vagy 65536 színváltozatot a három szín értékeinek keverésével. A '00' érték a színcsatornára nulla intenzitást jelent, miközben a fél, vagy a teljes intenzitást a '88' és az 'FF' jelenti. A tiszta kéket tehát a #0000FF jelenti, amíg a #FC8207 narancssárga színt fog adni.

A legtöbb számológép képes a különböző számrendszerek közötti konvertálásra.

Az így megjelölt megjegyzések könnyen kiszűrhetők mielőtt a kódot megpróbálnánk végrehajtani. Ehhez, a MY_DIALOG (vagy ahogy te elnevezted) változóban tárolt szöveget keresztülküldöd egy sed utasításon, ami eltávolítja a megjegyzéseket és tárolja a kimenetet a "comment-less" új változóban, készen a

normál használatra. Nem kell értened a sed használatát ahhoz, hogy az eljárást alkalmazd. Egyszerűen csak a következő példában látható sort add hozzá. Kíváncsiaknak, a sed lecserél (substitute – innen az s) bármilyen szöveget, ami a kezdő ## után álló szám, vagy karakter (.*) arra, ami a záró kettős törtjel (//) között található. Ebben az estben nincs ott semmi, ami a várt eredményt, a szöveg törlését eredményezi. A megjegyzések vagy egy külön sorban, vagy egy kódot tartalmazó sor végén jelennek meg, így a kód változatlanul megy át.

#!/bin/bash

##	A dupla kett s	kereszt megjegyzés		
##	a bash kódolású	szkriptekben is.		
##	De ott nem kell	eltávolítani		

MY_DIALOG='## gtkdialog kód itt indul <vbox>## Egy egyszer gomb megjelenítése<button ok=""></button>## Ezek a megjegyzések el lesznek távolítva</vbox> '## A gtkdialog kód vége
A sima kett s keresztek az XML kódon kívül szükségesek
Az XML kódból a megjegyzéseket ezzel a sorral távolítsd el # A MY_GUI-t és a MY_DIALOG a-ot cseréld le a kódodnak megfelel en
export MY_GUI="`echo "\$MY_DIALOG" sed 's/##.*//'`"
A gtkdialog meghívása az XML kód végrehajtására # átadva a módosított XML kódot # amib l a megjegyzések el lettek távolítva
gtkdialog -p MY_GUI
A megjegyzések az eredeti fájlban megmaradnak
Összehasonlításul íme ugyanez, a megjegyzések nélküli kód.
#!/bin/bash

MY_DIALOG=' <vbox> <button ok></button>

</vbox>'

gtkdialog -p MY_DIALOG

Az előző egyszerű példában a megjegyzések szükségtelenek, de egy sokkal összetettebb kód esetén segíthetik a hibakeresést, a módosítást és a karbantartást.

A színgomb és a forgatásgomb widgetek használata jelzésekkel

A színgomb widget a hasznos színválasztó párbeszéd megjelenítésére használatos, opcionális eszközsúgóval és egy pipettával, amivel a képernyőről bárhol lehet színt kiválasztani. A kiválasztott szín ezután a gomb színében jelenik meg és a szín értéke változóba kerül.

¥0	Choose a	color	_	_	/ 6	0
	Hue:	135	\$	Red:	13	\$
	Saturation:	94	\$	<u>G</u> reen:	223	^ V
	⊻alue:	87	0	<u>B</u> lue:	66	0
	Opacity:			-0	193	
	Color name:	#0D	DF42			
The previously-select comparison to the co	d color, for lor you're selecti	ing nov	". –	P <u>OK</u>	<u>C</u> anc	el

A színgomb widget alapul beállított megjelenése elég érdektelen, ahogy a lenti képen is látható.

Click to choose a color	1
Click to choose a color	J
≪ <u>O</u> K	

Javíthatunk ezen a gtkdialog jelzésnek nevezett tulajdonságával, amit a második cikkben már bemutattam. Ha kikapcsoljuk a színgomb widget láthatóságát, továbbra is aktív maradhat, de a párbeszédben nem jelenik meg. Így a párbeszédben elhelyezhetünk egy gombot, vagy vezérlőt és annak a jelét használjuk a színgomb működtetésére, azaz a színválasztó párbeszéd megjelenítésére.

Programozás Gtkdialog-gal, negyedik rész

A következő példában, a PCLinuxOS tárolójában is megtalálható, csodálatos és erőteljes ImageMagick eszköztá-rának sok funkciója közül az egyiket fogom használni.

Az ImageMagic logója egy kép, amin egy varázsló éppen varázsol. Szeret-ném a varázslót sokkal zöld-barátabbá változtatni azzal, hogy a szövegét és a palástját kékről zöldre változtatom.

Az ImageMagick-nak van egy mogrify (átalakítás) funkció, amivel mindenfélét megváltoztathatsz a képen, mint átlátszóságot és – amire most kíváncsiak vagyunk – a kitöltőszínt. Közelebbről megnéz-



ve a képet láthatjuk, hogy sok kék árnyék és színezet van és egyik sem igazi kék.

Ha megpróbáljuk kizárólag a kék pontokat megváltoztatni, akkor a mogrify hatására az összes teljesen kék színű pont új színt kap, miközben a többit nem változik. A mogrify 0 és 100 közötti értéket fogad el opcióként, meghatározandó a kiválasztás pontosságát. Az alacsonyabb érték sokkal pontosabb, amíg a magasabb értéknél kevésbé pontos, ún. "bolyhos", mintegy a közelítő, "valami ilyesmi" értéket vár.

Mivel nem tudjuk, mennyire "bolyhos", ezért módot kell adni az érték változtatására a határértékeken belül. E célból a spinbutton (görgető gomb) widget használata mellett döntöttem. Az utolsó párbeszéd különféle h- és vbox-

okat, illetve egy pixmap widget-et, egy reitett színgomb widget-et, egy – a

színgomb widget-nek jelzést küldő – díszítő gombot, egy spinbutton widget-et, egy változásokat visszavonó gombot, néhány címkét és egy – a változásokat elfogadó és a varázsló új öltözetének képéről másolatot mentő – OK gombot tartalmaz

Íme a kód:

#!/bin/bash

cp -f im-logo.png im-changed.png cp -f im-logo.png im-revert.png echo -n 50 > /tmp/SB1



export MY_GUI='<window title="Színgomb" resizable="false">

<vbox width-request="400" border-width="10">

<action>mogrify -fuzz `cat /tmp/SB1`"%"
-fill \$CB_1 -opaque "#0000ff"
im-changed.png</action>
</colorbutton>

<pixmap auto-refresh="true">

```
</hbox>
```

```
<hseparator></hseparator>
```

```
<button image-position="2">
```

```
<width>80</width>
<height>75</height>
<input file>revert.png</input>
<action function="command">
cp -f im-test.png im-changed.png
</action>
</button>
```

```
</hbox>
```

```
</text>
<text>
<label>Visszavonás</label>
</text>
</hbox>
<hseparator></hseparator>
<hbox homogeneous="true">
<button ok></button>
</hbox>
```

```
</vbox>
```

</window>

gtkdialog -p MY_GUI rm -f /tmp/SB1 rm -f im-revert.png

A munkához előbb az eredeti fájlról másolatot készítünk és beállítjuk a "bolyhosságot". A színgomb láthatósága kikapcsolt, felcímkéztük és a választott értéket a CB_1 változóba mentettük. Az alapértéket a varázsló öltözetéből, valahonnan középről vesszük, de ezt változtatjuk meg, amikor a színnek új értéket adunk. Az alfa értékét itt történetesen nem használjuk, de semmi bajt nem okoz, ha felvesszük ide. A színválasztó párbeszéd bezárásakor végrehajtott művelet a mogrify, ami a bolyhosság értékét a spinbutton állásához igazítja, a változtatni kívánt szín pedig a #0000ff – tiszta kék. A kimenet az egyik, általunk készített másolatba kerül.

Az első gomb widget az, amelyik a láthatatlan színgombnak aktiváló jelet küld. Amikor ezt a gombot lenyomjuk, button-press-event jelzés keletkezik és közöljük a gtkdialog-gal, hogy erre a jelzésre aktiválja a színgomb widget CB_1-et.

A görgető gomb tartományát és léptékét a címke attribútumánál határozzuk meg. 50, mint induló érték jónak tűnik. A leave-notify-event jelzés akkor keletkezik, amikor az egérmutató elhagyja a widget-et és az itt végrehajtandó művelet a görgető gomb aktuális értékének mentése az output címkepárnál meghatározott fájlba.

A visszavonás gomb egyszerűen rámásolja a változatlan fájlt a megváltoztatottra, így újra próbálkozhatunk. A végső OK gomb kilép a szkriptből és az imchanged.png fájlban hagyja az eredményt.

A szkript elhagyása előtt az időszaki fájlok törlődnek.

Az eredmények



Programozás Gtkdialog-gal, negyedik rész

A hscale és a vscale widget-ek

A következő példában az ImageMagick mellett maradván a mogrify parancs átméretezés és forgatás opcióit fogjuk használni. A kód zöme hasonlít az előzőhöz, kiegészítve megjegyzésekkel a műveletek dokumentálására, ám két új, "scale" widget funkciót vezet be. Egyik, a skála mellett jelölőt megjelenítő elem, háromféle adatot képes kezelni. Azt az értéket, ahol a skála mellett a jelölőnek meg kell jelennie, a skálához képesti helyzetét és a jelölő mellett opcionálisan megjelenítendő szöveget. Ezeket függőleges vonal karakter választja el egymástól.

A jelölő pozíció értéke 0, 1, 2, vagy 3 lehet, ezek bal, jobb, fent és lent értelműek. A kísérő szöveg tartalma pango jelölő címkékkel formázható. https://developer.gnome.org/pango/stable/PangoMarkupFormat.html. Ezt arra használtam, hogy a képet átméretező csúszka függőleges skáláján minimum értéknek a 10-et jelöljem meg. Ugyan a 0 értéket is elfogadja, de ez esetben ennek semmi értelme sincs. A span címkéi közötti betűközökök fontosak és a szöveg pozicionálására szolgálnak.

Íme a kód:

```
#!/bin/bash
# másolat készítése az eredeti fájlról
cp -f pic1.jpg mog1.jpg
MY DIALOG='<window title="GTKDIALOG hscale és vscale Widget-
ek" resizable="false">
<vbox> ## A központi ablak.
  <frame Image Manipulation>
    <hbox> ## Ez tartalmazza a két pixmap és a vscale widget-et
      <pixmap>
        <width>"237"</width>
        <height>"237"</height>
        <input file>pic1.jpg</input> ## Az eredeti kép
      </pixmap>
      <pixmap auto-refresh="true">
        <input file>mog1.jpg</input> ## A módosított kép
      </pixmap>
      <frame Re-size>
## Az átméretezést beállító csúszka
        <vscale width-request="200" height-request="400"
        range-min="10" range-max="200" range-step="1"
        value-pos="1" inverted="true">
            <default>100</default>
```

```
<variable>VS_1</variable>
```

```
## A mogrify parancs használja ezt az értéket
```

PCLinuxOS Magazine

```
<item>
         "10 | 1 |<span fqcolor='"'red'"'> 10% </span>"
   </item>
   <item>"20
               0 |<span> 20%</span>"</item>
   <item>"40
                  <span> 40% </span>"</item>
               1
   <item>"60
               0 |<span> 60%</span>"</item>
              1 |<span> 80% </span>"</item>
   <item>"80
               0 |<span> 100%</span>"</item>
   <item>"100
   <item>"120
               1 1 span> 120% 
               0 |<span> 140%</span>"</item>
   <item>"140
   <item>"160
               1 <span> 160% </span>"</item>
   <item>"180
                0 |<span> 180%</span>"</item>
              1 | 1 | <span> 200% </span>"</item>
   <item>"200
   <output file>/tmp/v out</output>
 </vscale>
</frame> ## A átméretezési rész vége
```

</hbox>

```
<frame Rotate>
```

```
## A forgatás szögét beállító csúszka
    <hscale width-request="400" height-request="120"</pre>
   range-min="0" range-max="360" range-step="1"
   value-pos="3" inverted="false">
     <default>0</default>
       <variable>HS 1</variable>
## A mogrify parancs használja ezt az értéket
     <item>"45 | 2 |<span> 45o</span>"</item>
     <item>"90 | 3 |<span> 90o</span>"</item>
     <item>"135 | 2 |<span> 135o/item>
     <item>"180
                  3 |<span> 180o</span>"</item>
     <item>"225
                 3 <span> 270o</span>"</item>
     <item>"270
     <item>"315 | 2 |<span> 315o</span>"</item>
     <item>"360 | 3 |<span> 360o</span>"</item>
    </hscale>
  </frame> ## A forgatási rész vége
    <pixmap auto-refresh="true">
## A kép frissítése, ha változott
     <input file>im-logo.jpg</input>
    </pixmap>
</hbox>
<hbox homogeneous="true">
    <button image-position="2">
## Kép mentése és kilépés
```

```
<input file stock="gtk-save"></input>
  <action>EXIT:ok</action>
</button>
```

<button image-position="2">

Változtatások visszavonása

</button>

<button image-position="2">

</hbox>

<hbox>

```
<button image-position="2">
     <label>"Megszakít"</label> ## Abandon all changes
     <input file stock="gtk-cancel"></input>
     <action>EXIT:cancel</action>
</button>
```

</hbox>

```
</frame> ## A képmanipulálási rész vége
</vbox>
</window>'
```

Megjegyzések eltávolítása
export MY_GUI="`echo "\$MY_DIALOG" | sed 's/##.*//'`"

A kód futtatása
gtkdialog -p MY_GUI

Takarítás
rm -f /tmp/h_out
rm -f /tmp/v_out





А

képet fekete keretbe tettem, hogy az átalakítások hatását bemutassam. Amikor a képet forgatod, látni fogod a négy fehér háromszöget a kép körül és egy kislányt, aki a kép méretének viszonylagos változását mutatja. A négy háromszög hozzáadódik és növeli a kép méretét, de a kislány mérete nem változik, hacsak nem használsz átméretezési effektust is.

PCLICUS Full Monty ...

FREE SOFTWARE

Everything you might want or need – plus the kitchen sink!

