libocelli - Feature #237

Implement icons

02.09.2022 18:13 - Maximilian Seesslen

Status:	Erledigt	**Beginn:**	02.09.2022
Priorität:	Normal	**Abgabedatum:**	
Zugewiesen an:	Maximilian Seesslen	**% erledigt:**	0%
Kategorie:		**Geschätzter Aufwand:**	0.00 Stunde
Zielversion:	v0.1.0	**Aufgewendete Zeit:**	0.00 Stunde
CS Zielversion:			

Beschreibung

Does Biwak already support such thing? Expect the drawLine.
Set background color; Alpha-Channel would be nice.
Its not an Widget.

Like Qt: button->setIcon(QIcon("open.xpm"));
48x48x4 9216 Bytes
only alpha:
48x48x1 2304 Bytes

Icons

https://www.iconfinder.com/search/icons?q=pretzel
https://www.iconfinder.com/search/icons?q=pretzel&price=free
git clone https://github.com/google/material-design-icons/

imagemagick (graphicsmagic provides GIF)

```
#!/bin/bash

set -e -u

mkdir -p 32
for ifile in *.png; do
    convert $ifile -define h:format=a -depth 1 -background white  -alpha off -resize 32x32 32/${ifile%.*}.h
    convert $ifile -define h:format=rgb -depth 1 -background white -alpha off -resize 32x32 32/${ifile%.*}.png

# convert $ifile -define h:format=rgb -depth 1 -background white -alpha deactivate -resize 32x32 32/${ifile%.*}.png
    convert $ifile -background white -alpha remove -flatten -alpha off -depth 1 -resize 32x32 32/${ifile%.*}.png
    convert 32/${ifile%.*}.png -monochrome 32/${ifile%.*}_1.png

done
```

There is an problem with the argument order in imagemagick. The examples above only work with pngs with alpha channels, not SVGs;
This works with SVGs:

```
convert -define h:format=a -depth 8 -background none -resize 48x48 $ifile 48/${ifile%.*}_alpha1.h
```

Historie

#1 - 02.09.2022 18:22 - Maximilian Seesslen

- Beschreibung aktualisiert

#2 - 02.09.2022 18:29 - Maximilian Seesslen

- Beschreibung aktualisiert

#3 - 02.09.2022 18:34 - Maximilian Seesslen

- Beschreibung aktualisiert

#4 - 05.09.2022 09:49 - Maximilian Seesslen

- Beschreibung aktualisiert

#5 - 06.09.2022 16:24 - Maximilian Seesslen

One channel:
FG: 0xFF
BG: 0x10

fAlpha: 0xFF
bAlpha: (0xFF-0xFF) = 0x0
nFGg= (0xFF*0xFF)/0xFF = 0xFF
nBG= (0x10*0x0)/0xFF = 0X00

fAlpha: 0x0
bAlpha: (0xFF - 0x00) 0xFF
nFGg= (0xFF*0x0)/0xFF = 0x0
nBG= (0x10*0xFF)/0xFF = 0X10

One channel:
FG: 0x20
BG: 0x10
fAlpha = 0x7f
bAlpha = 0x80
nFGg= (0x20*0x7F)/0xFF = 0xF
nBG= (0x10*0x80)/0xFF = 0x8; 0xF+0x8=0x17

```
uint8_t blend(fg, bg, alpha)
{
   return  ( ( fg * alpha ) / 0xFF ) + ( ( bg * ( 0xFF - alpha ) ) / 0xFF )
}

assert( 0xFF, 0x00, 0xFF) == 0xFF
assert( 0xFF, 0x10, 0x00) == 0x10
```

#6 - 06.09.2022 20:53 - Maximilian Seesslen

- Beschreibung aktualisiert

#7 - 07.09.2022 10:05 - Maximilian Seesslen

- Beschreibung aktualisiert

#8 - 19.09.2022 22:20 - Maximilian Seesslen

- Status wurde von Neu zu Erledigt geändert