

OLUG - Gimp Presentation

September 6th, 2005

Eric Pierce (e.omaha@gmail.com)

1. Introduction

1. What is the Gimp? www.gimp.org
 1. Raster image editing software – edit digital image w/pixel precision
 2. Been around since mid 90s
 3. Installed by default on just about any desktop oriented Linux install
 4. Also available for Macs and Windows
2. What I'm going to talk about – epierce.freeshell.org/gimpfiles.tgz
 1. Photo touch up - Red eye removal
 2. Photo touch up - Removing color casts
 3. Project - “Paint your walls before you paint them”
 4. Script-fu Primer for Adam

2. Gimp's GUI customization

1. Drag/Add: tool options, brushes, fonts, document history, layer, channel, path, undo history
2. Copy keyboard shortcuts – **Mention that they may differ from default!**
3. Pref: Ants to 75; toolbox->show active image; Appearance

3. Photo Editing - Red eye removal from photos

1. Selection tools/techniques - It's all in the selection
 1. Fuzzy Select
 1. Threshold setting (20)
2. Feather – do 1 pixel on left and 2 pixel on the right
3. Channels – unselect green and blue
4. Desaturate or Hue/Saturation to remove red
5. Channels – reselect blue

4. Photo Editing - Removing color casts from photos

1. *Method 1* – Levels tool
2. *Method 2* - Filter Pack
 1. Used when it's hard to find a solid white/gray/black.

5. Project – “Paint your walls before you paint them”

1. Explain approach/steps to problem
2. Make selection of the walls
 1. Use paths to create a “mask” (not a true Gimp mask)
 2. Movement/zoom techniques – middle button, (ctrl)pageUp/pageDown, +/-
 3. Get Julia involved
3. Color to alpha
4. Show color(s) for walls on a background layer
5. Other scenarios
 1. What if your walls aren't white to begin with?
 1. Desaturate and lighten walls appropriately
 2. What if you have wallpaper up?
 1. You're screwed

6. Script-fu Primer

1. Introduction

1. Macro similarities and differences
 1. Script-fu based on Scheme language
 1. Control structures
 2. Condition checking (if/then)

2. Advantages of Script-fu

1. Not only language for writing plug-ins
 1. Also bindings for Python and Perl
 2. Native plug ins in C
2. Nothing to install / portability between OSes

3. Getting started - Script-fu tools

1. Script-fu Console – everything on one line
2. Gimp's Procedure Database – Careful when copying procedure name.
 1. Procedures = a functionality of the Gimp
3. *Refresh Scripts

4. Scheme characteristics/quickies

1. Equations are in prefix notation
 1. **(+ 3(* 5 2))** (Note: need a space between operand and operator)
2. Lists/Accessing elements in a list – ex. **'(1 2 3)**
 1. **car** (get 1st element of a list) – Ex. **(car '(1 2 3)) = 1**
 2. **cadr** (get 2nd element of a list) – Ex. **(cadr '(1 2 3)) = 2**
 3. Why are lists important in Script-fi
3. Semi-colon for commenting

5. Overview of a script's parts – pull up olug.scm

***Image vs Toolbox

1. Script registration with the Gimp (put somewhere convenient while testing)
 1. Find new script in the menu and in the procedure database
 2. underline is the mnemonic shortcut key
 3. Color spaces: **RGB, RGBA, GRAY, GRAYA, INDEXED, INDEXEDA, ***
 4. Arguments used: **SF-IMAGE; SF-DRAWABLE; SF-ADJUSTMENT; SF-COLOR**
 1. Reference **test-sphere.scm**
2. Main function block
 1. Variable Initialization block
 1. **(let* ())** outer block encompasses entire function block
 2. Use **set!** outside of let inner block for

***run script once to show steps in drop shadow creation

3. Main code block
 1. Mostly simple procedure calls to go through the steps to create shadow

6. Debugging

1. **(gimp-message msg) / (gimp-message (number->string n))**

7. Finalizing

1. **(gimp-image-undo-group-start theImage) / (gimp-image-undo-group-end theImage)**

8. Script-fu References

1. docs.gimp.org
2. Installed Script-fu scripts are an excellent reference

9. Future: Tiny-fu to replace Script-fu

1. based on TinyScheme
2. regex pattern matching
3. date & time functionality