

LZW Compression and Image Files

Contents

LZW Compression.....	3
Basic Terms for GIF and PNG.....	3
Goal: Deciding Whether to Use a GIF or PNG file.....	3
How to Create a GIF using the Command Line on a OS X Operating System.....	4
Converting to a PNG File using Command Line on Mac.....	4

LZW Compression

A brief explanation of LZW Compression

Lempel-Ziv-Welch (LZW) is a lossless data compression algorithm that was first published by Terry Welch in 1984 as a modification of the LZ77 and LZ78 compression algorithms developed by Jacob Ziv and Abraham Lempel. It is a dictionary based compression algorithm that is relatively simple for developers to implement and is widely used in GIF file formats.

LZW is a dictionary-based compression method that takes advantage of repetitive patterns. LZW is based on a standard character set dictionary of 256 characters. To decompress the data, LZW does not require the dictionary assignments that were featured in the compression; this is because both the encoding and the decoding programs utilize the same 256 characters. To decompress, LZW reads an assigned character and “looks up” the character in the dictionary to see what substring it was applied to. This process is repeated until the information is decoded.

LZW is used to compress GIF files by removing horizontal redundancies. This caused a few issues in the early 1990's, when the patent holders for LZW began collecting licensing fees on the use of the algorithm. These issues inspired a Usenet.com.graphics discussion and resulted in the creation of the PNG file. PNG files use a variation of LZW compression that does not fall under the Unisys patent.

Basic Terms for GIF and PNG

Some basic terms involved in the compression of GIF and PNG files.

Term	Definition
LZW Compression	a lossless data compression algorithm that was first published by Terry Welch in 1984 as a modification of the LZ77 and LZ78 compression algorithms developed by Jacob Ziv and Abraham Lempel
Lossless Data Compression	data compression algorithm where original data can be perfectly reconstructed from the compressed data
Dictionary Coder	type of lossless data compression algorithms that enable a table/ dictionary to encode and decode files.
LZW Table/ Dictionary	a dictionary developed by scanning a text for repeated letters and letter strings and giving them a corresponding "dictionary" entry of bits

Goal: Deciding Whether to Use a GIF or PNG file

This task will aid you in selecting the GIF or PNG file format.

Have a webpage that is currently in development

Have the appropriate software and hardware

Have a goal in mind for what you would like your image file to achieve.

1. Ask yourself: does this need to be supported on older browsers? If so, mark 1 for GIF.
2. Ask yourself: does this need to be an animated file? If so, mark 1 for GIF.

3. Ask yourself: is this a logo? If so, mark 1 for PNG.
4. Ask yourself: does this require transparency? If so, mark 1 for PNG.
5. Ask yourself: is this line art? If so, mark 1 for PNG.

How to Create a GIF using the Command Line on a OS X Operating System

This task will enable you to create a GIF using the terminal on a Mac computer.

Ensure that you meet all of the software and hardware requirements.

Have an operational understanding of how a terminal works.

Select a video clip.

Download mplayer, imagemagick, and gifsicle. All of these can be downloaded from the Homebrew Package Manager

Have a "gif" directory

This task will enable you to create a GIF using the terminal on a Mac computer.

1. Take the segment of video that you chose and note the start in H:MM:SS timestamp format. Note the end time as well.
2. Use the mplayer package to transform the video into a series of images.
3. Input into the terminal: "mplayer -ao null -ss H:MM:SS-endpos 5 -vo gif89a:outdir=gif videofile.mp4 " ao= audio output, it is null because the gif will be silent; ss is the start position, input the time stamp in H:MM:SS format; endpos is the end position, input the number of seconds that elapse from the start time stamp to where you would like the gif to end; gif89a is the output format; outdir=gif is our output director; videofile.mp4 if the format of the video we have trimmed.
4. You should now have still images in your "Gif" directory. Go through and remove the unnecessary images at the start and end, so the gif appears smooth. If you want, you can manually do this outside of the command line by going into the directory.
5. Use the gifsicle package to convert the series of images into an animated " gif. gifsicle --colors=256 --delay=4 --loopcount=0 --dither -O3 *.gif " animation.gif. 256 is a good number of colors of an animated gif to still be easily uploaded; loopcount is the number of times the gif repeats itself; dither adds noise to the image file to randomize quantization error/ prevent color banding

Converting to a PNG File using Command Line on Mac

This task will help you convert various file formats to a PNG file on a Mac operating system.

Have a webpage that is currently in developmnet

Have the appropriate software and hardware

Have an OS X operating system.

Have a file that you want to convert to PNG

Connect to the directory your image is located in

If converting multiple images, have a sub file called "converted".

1. Open the terminal. We will be using "sips". sips is Apple's pre-installed "simple image processing system".
2. Type: "sips -s format png test.jpg --out test.png" into the terminal; Here we are converting from a jpg file.

3. If you wish to convert from another file (ie, a pdf) replace ".jpg" with ".pdf".
4. To convert multiple image files at once, type "for i in *.jpeg; do sips -s format png \$i --out Converted/\$i.png;done" Here we've converted all "jpeg" files within our directory to png and placed them in the "converted" folder.
5. If you wish to convert your batch of images from another file (ie, a pdf) replace ".jpg" with ".pdf".
6. In the event that a "libpng warning" pops up on your terminal, you can ignore it. It is an issue related to different versions of image control programs on the same computer.