# Bhujbal Knowledge City

# MET's Institute Of Engineering

**Adgaon, Nashik-422003.**

**ACADEMIC YEAR 2010-2011.**

**A**
**SEMINAR**
**ON**

# "Graphics Interchange Format Animation"

**SUBMITTED IN**
**DEPARTMENT OF MCA**

**GUIDED BY**
Prof. Apeksha Gawande

**PRSENTED BY**
Mr. Jagdish Chopde

**SYMCA**
**ROLL NO - 05**

# Bhujbal Knowledge City

# MET’s Institute Of Engineering

**Adgaon,Nashik-422003.**

Affiliated to

## University Of Pune

Department Of Master of Computer Application

## *CERTIFICATE*

**This is to certify that seminar entitled**

**”Graphics Interchange Format Animation”**

**Has satisfactorily completed by**

**Mr. Jagdish Chopde**

**In partial fulfillment of term work for**

**Degree in Master of Computer Application**

**For the academic year 2010-2011**

(Prof.Apeksha Gawande)
Seminar Guide

(Prof.Aruna Deogire)
H.O.D.

# Acknowledgements

Every work is source which requires support from many people and areas. It gives me proud privilege to complete the seminar on *"Graphics Interchange Format Animation"* under valuable guidance and encouragement of my guide **Prof. Apeksha Gawande**.

I am also extremely grateful to our respected H.O.D.(MCA Dept.) **Prof.Aruna Deogire** and **Prof.Swapnil Kulkarni**(Seminar Coordinator) for providing all facilities and every help for smooth progress of seminar.

I would also like to thank all the Staff Members of MCA Department for timely help and inspiration for completion of the seminar.

At last I would like to thank all the unseen authors of various articles on the Internet, helping me become aware of the research currently ongoing in this field and all my colleagues for providing help and support in my work.

Mr. Jagdish Chopde.
SYMCA

# Abstract

The Graphics Interchange Format (GIF) is a bitmap image format that was introduced by CompuServe in 1987 and has since come into widespread usage on the World Wide Web due to its wide support and portability.

The format supports up to 8 bits per pixel thus allowing a single image to reference a palette of up to 256 distinct colors. The colors are chosen from the 24-bit RGB color space. It also supports animations and allows a separate palette of 256 colors for each frame. The color limitation makes the GIF format unsuitable for reproducing color photographs and other images with continuous color, but it is well-suited for simpler images such as graphics or logos with solid areas of color.

GIF images are compressed using the Lempel-Ziv-Welch (LZW) lossless data compression technique to reduce the file size without degrading the visual quality. This compression technique was patented in 1985. Controversy over the licensing agreement between the patent holder, Unisys, and CompuServe in 1994 spurred the development of the Portable Network Graphics (PNG) standard; since then all the relevant patents have expired. If multiple images are stored within a GIF file, they can be viewed sequentially much like a slide show or a small animated movie. The way they appear is defined by control extensions built into the file. With the appropriate tools, such as Ulead GIF Animator.

# Contents

# List of Figures

# Chapter 1

# Introduction

The Graphics Interchange Format (GIF) is a bitmap image format that was introduced by CompuServe in 1987 and has since come into widespread usage on the World Wide Web due to its wide support and portability.

The format supports up to 8 bits per pixel thus allowing a single image to reference a palette of up to 256 distinct colors. The colors are chosen from the 24-bit RGB color space. It also supports animations and allows a separate palette of 256 colors for each frame. The color limitation makes the GIF format unsuitable for reproducing color photographs and other images with continuous color, but it is well-suited for simpler images such as graphics or logos with solid areas of color.

GIF images are compressed using the Lempel-Ziv-Welch (LZW) lossless data compression technique to reduce the file size without degrading the visual quality. This compression technique was patented in 1985. Controversy over the licensing agreement between the patent holder, Unisys, and CompuServe in 1994 spurred the development of the Portable Network Graphics (PNG) standard; since then all the relevant patents have expired.

If multiple images are stored within a GIF file, they can be viewed sequentially much like a slide show or a small animated movie. The way they appear is defined by control extensions built into the file. With the appropriate tools, such as Ulead GIF Animator.

# Chapter 2

# History of GIF

CompuServe introduced the GIF format in 1987 to provide a color image format for their file downloading areas, replacing their earlier run-length encoding (RLE) format, which was black and white only. GIF became popular because it used LZW data compression, which was more efficient than the run-length encoding that formats such as PCX and MacPaint used, and fairly large images could therefore be downloaded in a reasonably short time, even with very slow modems.

The original version of the GIF format was called 87a. In 1989, CompuServe devised an enhanced version, called 89a, that added support for animation delays (multiple images in a stream were already supported in 87a), transparent background colors, and storage of application-specific metadata. The 89a specification also supports incorporating text labels as text (not embedding them in the graphical data), but as there is little control over display fonts, this feature is not widely used. The two versions can be distinguished by looking at the first six bytesof the file (the ”magic number” or ”signature”), which, when interpreted as ASCII, read ”GIF87a” and ”GIF89a”, respectively.

GIF was one of the first two image formats commonly used on Web sites, the other being the black and white XBM JPEG came later with the Mosaic browser. The feature of storing multiple images in one file, accompanied by control data, is used extensively on the Web to produce simple animations.

# Chapter 3

# Working of GIF Animation

## 3.1 Animation Basics

Animation is created by displaying a series of overlapping images (cels) in rapid succession. Each cel's image differs only slightly from the ones preceding and following it. When enough of these cels are grouped together, the differences between them create the illusion of motion when played. In traditional animation, cels are drawn and painted by hand, although in recent years the animation industry has been moving away from traditional animation and towards computer-assisted animation. While the concepts used to create the animation are essentially the same, the methods used differ greatly. The key defining element of all animation, and of movies in general, is the number of cels used to create the animation. The more cels a single animation contains, the smoother the motion of animated characters and objects appear to be. The less cels used, the more herky-jerky their movements will appear. In traditional animation, the number of cels becomes the length of the animation in seconds. In computer animation, however, the number of cels contributes to the ultimate size of the animation file. When creating animation on the computer, a balance must be found between animation quality and file size, both of which are affected by the cel count of the animation. Defining the best trade-off point is never easy, but with careful planning and foresight, you can create captivating animations that don't require unreasonable amounts of disk space.

## 3.2 How GIF Animation Works

GIF animation takes advantage of the GIF format's built in capabilities to store and display more than one image file. Unlike conventional film where the speed of the animation is defined by the speed of the film, these images each have an associated set of controls that dictate how, when, and for how long the image is displayed on the screen. Also unlike conventional film, the images contained within an animated GIF need not be the same size. Each image layer, or cel, can be any size you want, and can be positioned where you want regardless of the locations of the other cels in your animation.

The animation produced by the GIF file is produced in exactly the same manner as the method used in film: motion is created by rapidly displaying a sequence of similar images. While the methods used to produce this effect in both film-making and GIF animation differ greatly, the end result is the same - still images are brought to life before your eyes.

# Chapter 4

# GIF File Format

## 4.1 File Format

GIF files start with a fixed-length header (”GIF87a” or ”GIF89a”) giving the version, followed by a fixed-length Logical Screen Descriptor giving the size and other characteristics of the canvas. The screen descriptor may also specify the presence and size of a Global Color Table, which follows next if present.

Thereafter, the file is divided into segments, each introduced by a 1-byte sentinel:

- An image (introduced by 0x2C, a comma ’,’)
- An extension block (introduced by 0x21, an exclamation point ’!’)
- The trailer (a single byte of value 0x3B, a semi-colon ’;’), which should be the last byte of the file.

An image starts with a fixed-length Image Descriptor, which may specify the presence and size of a Local Color Table (which follows next if present). The image data follow: one byte giving the bit width of the unencoded symbols (which must be at least 2 bits wide, even for bi-color images), followed by a linked list of sub-blocks containing the LZW-encoded data.

Extension blocks (blocks that ”extend” the 87a definition via a mechanism already defined in the 87a spec) consist of the sentinel, an additional byte specifying the type of extension, and a linked list of sub-blocks with the extension data. Extension blocks that modify an image (like the Graphic Control Extension that specifies the optional animation delay time and optional transparent background color) must immediately precede the segment with the image they refer to.

The linked lists used by the image data and the extension blocks consist of series of sub-blocks, each sub-block beginning with a byte giving the number of subsequent data bytes in the sub-block (1 to 255), the series terminated by the empty sub-block (a 0 byte).

This structure allows the file to be parsed even if not all parts are understood. A GIF marked 87a may contain extension blocks; the intent is that a decoder can read and display the file without the features covered in extensions it does not understand. The full detail of the file format is covered in the GIF specification.

## 4.2 Palettes

GIF is palette-based: the colors used in an image (a frame) in the file have their RGB values defined in a palette table that can hold up to 256 entries, and the data for the image refer to the colors by their indexes (0-255) in the palette table. The color definitions in the palette can be drawn from a color space of millions of shades (224 shades, 8 bits for each primary), but the maximum number of colors a frame can use is 256. This limitation seemed reasonable when GIF was developed because few people could afford the hardware to display more colors simultaneously. Simple graphics, line drawings, cartoons, and grey-scale photographs typically need fewer than 256 colors.

As a further refinement, each frame can designate one index as a ”transparent background color”: any pixel assigned this index takes on the color of the pixel in the same position from the background, which may have been determined by a previous frame of animation.

Many techniques, collectively called dithering, have been developed to approximate a wider range of colors with a small color palette by using pixels of two or more colors to approximate in-between colors. These techniques sacrifice spatial resolution to approximate deeper color resolution. While not part of the GIF specification, dithering can of course be used in images subsequently encoded as GIF images. This is often not an ideal solution for GIF images, both because the loss of spatial resolution typically makes an image look fuzzy on the screen, and because the dithering patterns often interfere with the compressibility of the image data, working against GIF's main purpose.

In the early days of graphical web browsers, graphics cards with 8-bit buffers (allowing only 256 colors) were common and it was fairly common to make GIF images using the websafe palette. This ensured predictable display, but severely limited the choice of colors. Now that 32-bit graphics cards, which support 24-bit color, are the norm, palettes can be populated with the optimum colors for individual images. A small color table may suffice for small images, and keeping the color table small allows the file to be downloaded faster. Both the 87a and 89a specifications allow color tables of 2n colors for any n from 1 through 8. Most graphics applications will read and display GIF images with any of these table sizes; but some do not support all sizes when creating images. Tables of 2, 16, and 256 colors are widely supported.

## 4.3 True Color

There are at least two rarely-used methods that can generate a GIF that, if decoded according to the GIF89a standard, will produce an animation that ends with a 24-bit RGB true color image.

GIF89a was designed based on the principle of rendering images (known as frames when used for animation) to a logical, fixed-size screen. Each image could optionally have no delay after it is rendered, and could have its own 256-color palette. Also, each image need not fill the entire logical screen, and the animation can cease after the last frame; it need not begin again. The multi-frame, zero-delay, and unique-palette features, optionally combined with transparency, allow for each image to replace only a portion of the previous image's pixel data. When used without looping, a more-than-256-color final result can be achieved.

For example, a GIF can be encoded to render as a series of overlapping full-screen images, each image filling in color that wasn't in the previous one. Transparent pixels can be used to preserve colors from previous images. A similar method that does not use transparency is to encode the GIF to render as a series of less-than-full-screen images adjacent to each other, rather than overlapping.

## 4.4 Example GIF File (frames)

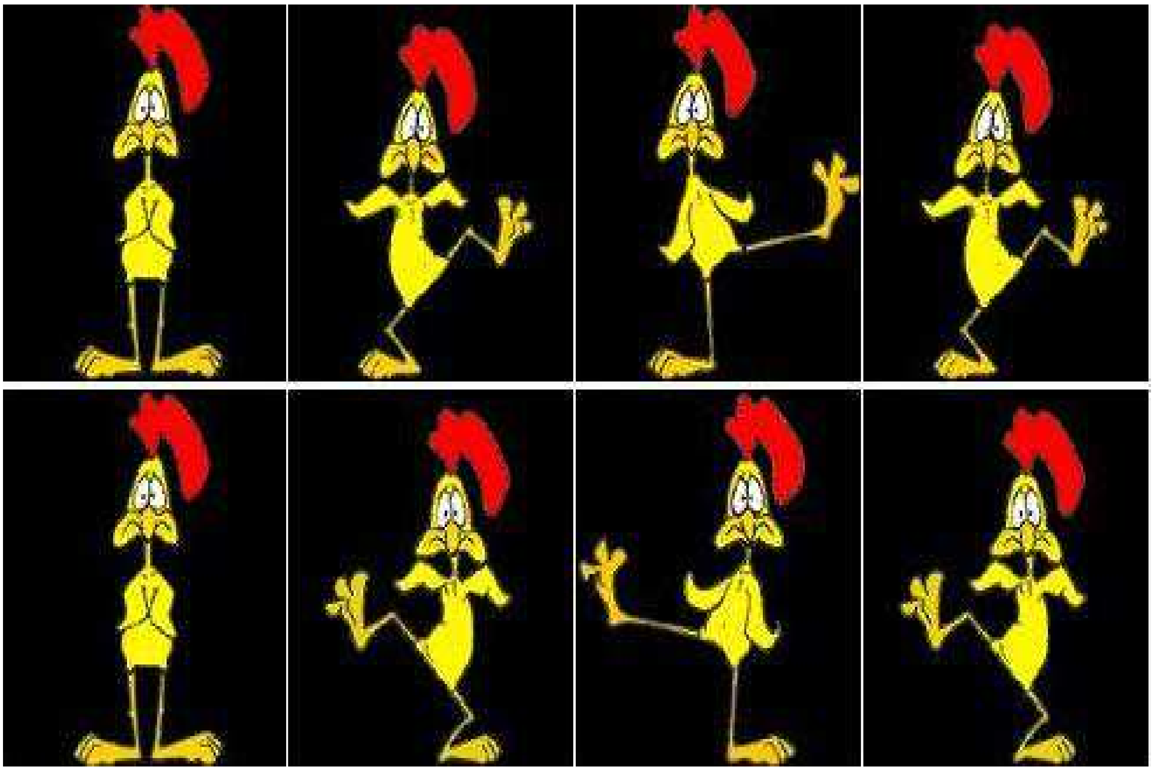
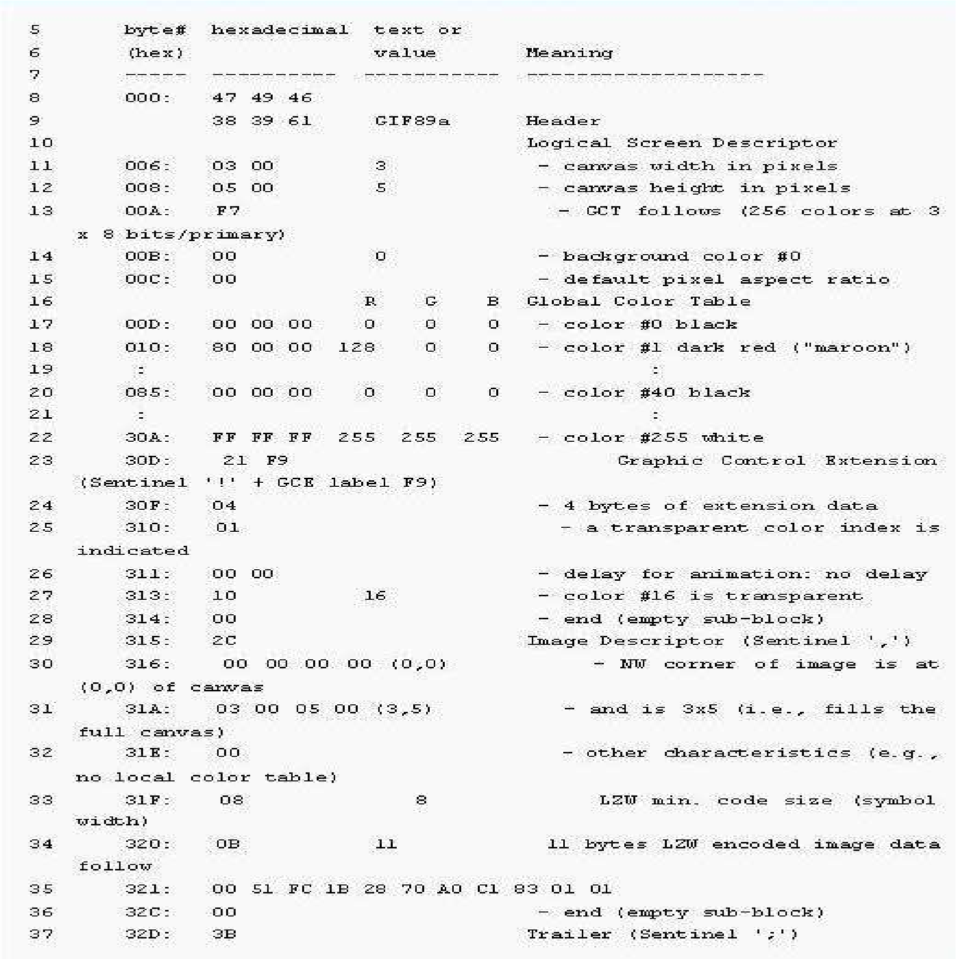


Figure 4.1: example of GIF Image (Animation)

## 4.5 Example GIF File (Codes)



```
byte#  hexadecimal  text or
(hex)               value         Meaning
-----  -----------  ------------  --------------------
000:   47 49 46
       38 39 61     GIF89a        Header
                                  Logical Screen Descriptor
006:   03 00        3              - canvas width in pixels
008:   05 00        5              - canvas height in pixels
00A:   F7                            - GCT follows (256 colors at 3 x 8 bits/primary)
00B:   00           0              - background color #0
00C:   00                          - default pixel aspect ratio
                    R    G    B   Global Color Table
00D:   00 00 00     0    0    0    - color #0 black
010:   80 00 00   128    0    0    - color #1 dark red ("maroon")
  :                                         :
085:   00 00 00     0    0    0    - color #40 black
  :                                         :
30A:   FF FF FF   255  255  255    - color #255 white
30D:    21 F9                              Graphic Control Extension (Sentinel '!' + GCE label F9)
30F:   04                          - 4 bytes of extension data
310:   01                            - a transparent color index is indicated
311:   00 00                       - delay for animation: no delay
313:   10           16             - color #16 is transparent
314:   00                          - end (empty sub-block)
315:   2C                         Image Descriptor (Sentinel ',')
316:    00 00 00 00 (0,0)                - NW corner of image is at (0,0) of canvas
31A:   03 00 05 00 (3,5)              - and is 3x5 (i.e., fills the full canvas)
31E:   00                             - other characteristics (e.g., no local color table)
31F:   08              8                  LZW min. code size (symbol width)
320:   0B            11             11 bytes LZW encoded image data follow
321:   00 51 FC 1B 28 70 A0 C1 83 01 01
32C:   00                          - end (empty sub-block)
32D:   3B                         Trailer (Sentinel ';')
```

Figure 4.2: GIF File Codes

# Chapter 5

# Ulead GIF Animator

## 5.1 About Software

Ulead GIF Animator is a historical computer software program for Microsoft Windows to create simple animated GIF files based on the GIF89a file format. It was freely downloadable from the Ulead Download Center but is now only available on third party download sites. It was also bundled with Microsoft Image Composer and Other related Softwares. Animations can be looped, spun, and faded in and out; users can set the size and transparency of images. Automatic and custom palette creation are supported.

## 5.2 Software Overview

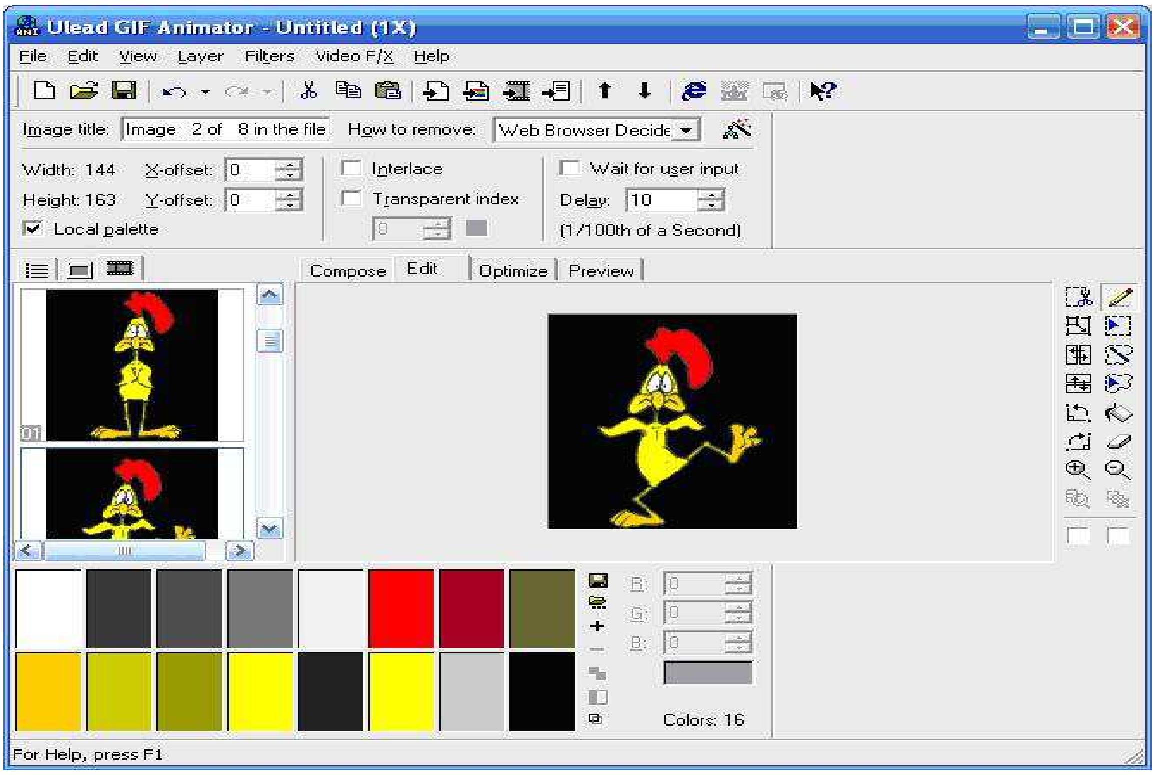


Image title: Image 2 of 8 in the file

How to remove: Web Browser Decide

Width: 144 X-offset: 0

Height: 163 Y-offset: 0

☑ Local palette

□ Interlace

□ Transparent index

□ Wait for user input

Delay: 10

(1/100th of a Second)

R: 0

G: 0

B: 0

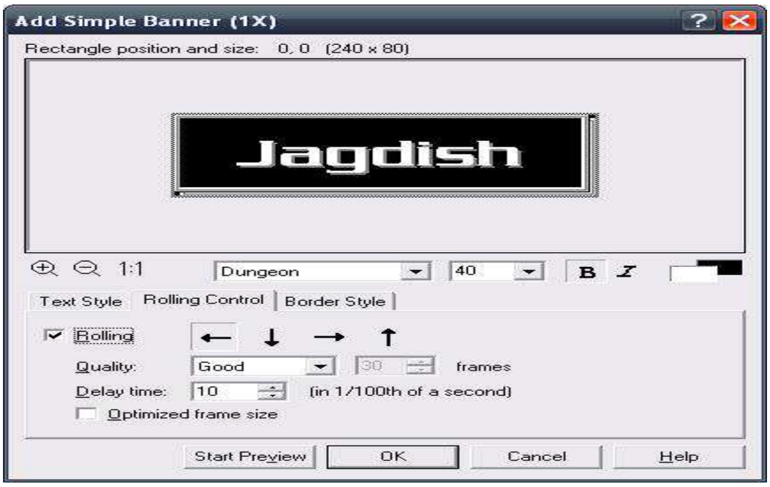
Colors: 16

For Help, press F1

Figure 5.1: GIF Animator 4.0 SnapShot

## 5.3 Adding simple banner text

This allows you to create scrolling text banners anywhere within your animation. These banners are additional frames that can be positioned however you like in relation to the rest of the animation. Keep in mind that these frames will increase the overall size of your animation file.
To add simple banner text to your animation:



Add Simple Banner (1X)

Rectangle position and size: 0, 0 (240 x 80)

☑ Rolling

Quality: Good

30 frames

Delay time: 10 (in 1/100th of a second)

□ Optimized frame size

Figure 5.2: simple banner text

1. Select the Layer: Add Banner Text - Simple command to open the Add Simple Banner dialog box.

2. Specify the font, size, and color of the text, as well as the default background color, by using the formatting controls located beneath the preview window.

3. Enter the contents of your banner text in the Text Style tab, in the Banner text box. Select the Merge with background image option to create banner text that blends in with your animation, otherwise the text frame will use the background color specified in the formatting controls below the preview window.

4. In the Rolling Control tab specify the direction in which text should scroll, as well as the quality and speed of the animation. By selecting the Optimized frame size option, you reduce file size considerably through shortening the width of the banner, but might limit you later when you want to edit the text in the image layer.

5. Switch to the Border Style tab and set the size and color of the banner text border. You can also add shadow to the border.

6. Click OK.

Note: If you create banner text with a color background and then manually set that color to be transparent, the banner text frames won't display properly unless you also set the removal option for each text layer as To previous state, a method not supported by all web browsers.

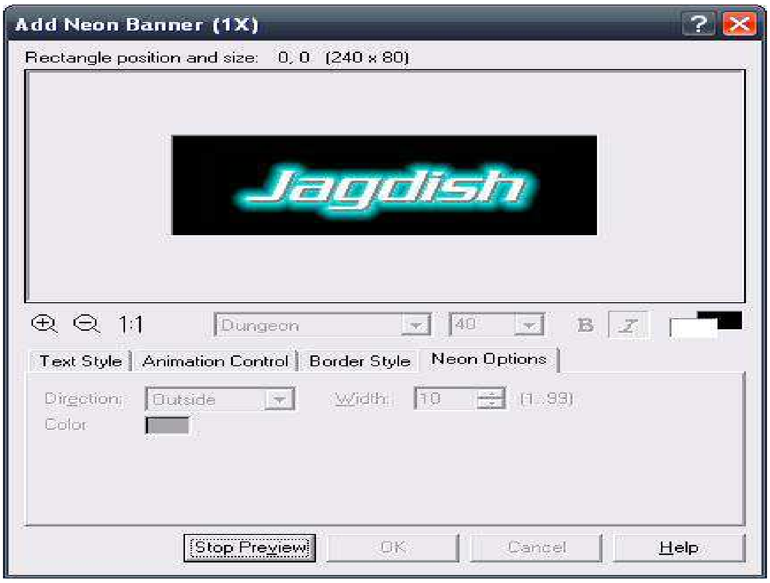
## 5.4 Adding a neon banner

Let your text radiate with an eye-catching neon banner. In addition to setting the size and color of the neon glow, you can also control the speed at which the neon glow radiates.
To add a neon banner to your animation:

1. Select the Layer: Add Banner Text - Neon command to open the Add Neon Banner dialog box.

2. Specify the font, size, and color of the text, as well as the default background color, by using the formatting controls located just beneath the preview window.

3. Enter the contents of your banner text in the Text Style tab, in the Banner text box. Select the Merge with background image option to create banner text that blends in with your animation, otherwise the text frame will use the background color specified in the formatting controls below the preview window.



Rectangle position and size: 0, 0 (240 x 80)

Direction: Outside

Width: 10 (1..99)

Figure 5.3: neon banner text

4. In the Animation control tab specify the quality and speed of the animation. By selecting the Optimized frame size option, you reduce file size considerably through shortening the width of the banner, but it might limit you later when you want to edit the text in the image layer.

5. Switch to the Border Style tab and set the size and color of the banner text border. You can also add shadow to the border.

6. Switch to the Neon Options tab and set how you want the banner text color to change during the course of the animation.

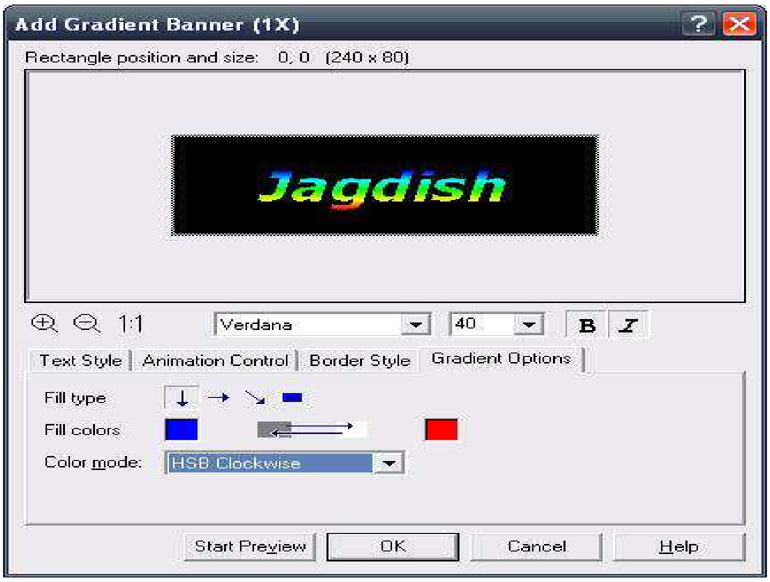
7. Click OK.

Note: If you create banner text with a color background and then manually set that color to be transparent, the banner text frames won't display properly unless you also set the removal option for each text layer as To previous state, a method not supported by all web browsers.

## 5.5 Adding a gradient banner

Gradient effects can make your banner text change its hue like a chameleon. Among many other options, you can make the colors shift starting from the center of the text for an radiating effect.

To add a gradient banner to your animation:



Rectangle position and size: 0, 0 (240 x 80)

Verdana 40

Fill type

Fill colors

Color mode: HSB Clockwise

Figure 5.4: gradient banner text

1. Select the Layer: Add Banner Text - Gradient command to open the Add Gradient Banner dialog box.

2. Specify the font, size, and color of the text, as well as the default background color, by using the formatting controls located just beneath the preview window.

3. Enter the contents of your banner text in the Text Style tab, in the Banner text box. Select the Merge with background image option to create banner text that blends in with your animation, otherwise the text frame will use the background color specified in the formatting controls below the preview window.

4. In the Animation control tab specify the quality and speed of the animation. By selecting the Optimized frame size option, you reduce file size considerably through shortening the width of the banner, but might limit you later when you want to edit the text in the image layer.

5. Switch to the Border Style tab and set the size and color of the banner text border.

6. Switch to the Gradient Options tab and set how you want the banner text color to change during the course of the animation.

7. Click OK.

Note: If you create banner text with a color background and then manually set that color to be transparent, the banner text frames won't display properly unless you also set the removal option for each text layer as To previous state, a method not supported by all web browsers.

## 5.6 Adding marquee text

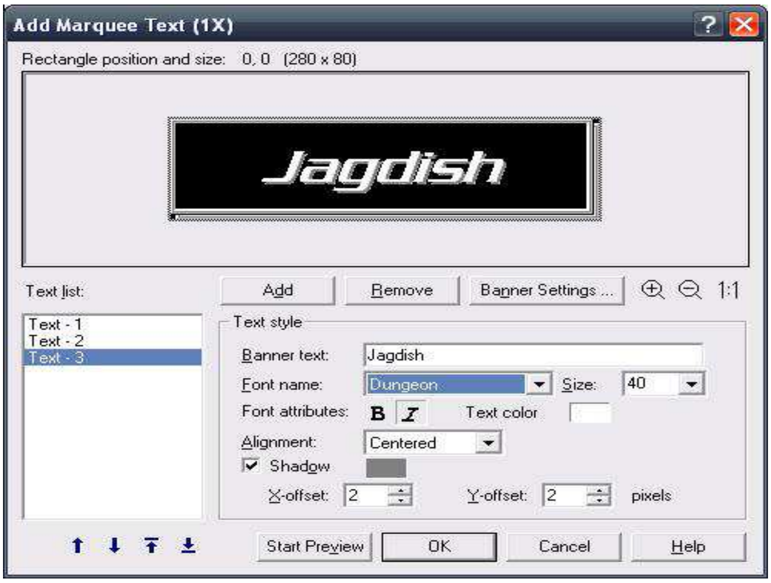
Applying the Marquee effect means that you can create many separate text strings that scroll across the background individually. Banner Settings also let you determine the Key frame delay, so that each text string can pause for a moment before making way for the next.

To add Marquee Text:

1. In the Text list box, specify the number of text strings that you want to use. Click Add to create additional text strings. To delete an unwanted text string from the list, select it and then click Remove.

2. After you've created your text list, select a specific text string in the list, then enter the desired text in the Banner text field.

3. In the Text style group, specify the attributes of your text string, including the shadow, color and alignment.



Rectangle position and size: 0, 0 (280 x 80)

Text list:

- Text - 1
- Text - 2
- Text - 3

Banner text: Jagdish

Font name: Dungeon   Size: 40

Alignment: Centered

☑ Shadow

X-offset: 2   Y-offset: 2 pixels

Figure 5.5: marquee banner text

4. Once you've finished entering the text, simply select another text string from the Text list box, and repeat steps 2 and 3. Each text string will retain only those attributes you assign to it, meaning that you can have different styles for different strings.

5. Click the Banner Settings button to specify Rolling control and Border settings. These are applied to all text strings. For more on Banner Settings, click here .

6. Click the Preview button to catch a preliminary glimpse of your banner, then click OK to apply the settings.

# Chapter 6

# References

- Ulead GIF Animator 4.0 Product of Ulead Corporation 1992-1999.
- Wekipedia - Graphics Interchange Format (History of GIF Technology) - 2010