

一.概述

GIF图像是有CompuServe公司开发的图形文件格式，GIF图像是基于颜色列表的（即GIF文件中存储的数据为该点的颜色在颜色列表中的索引值），同时GIF最多支持8位（即256色）的颜色展示。

二.GIF文件结构

基础知识

GIF文件内部是按块来进行划分的，其中包括控制块（Control Block）和数据块（Data Sub-blcks）两种。

- 控制块是用来控制数据块行为的，不同的控制块一般包含不同的控制参数。
- 数据块是用来记载数据的，数据块是一组8-bit组成的数据流，其中每个数据块的大小最大为**256**字节，其中第一个字节用于指出该数据块所含字节数，即一个数据块记录数据所用字节最多为**255**字节。以下是数据块的结构：

BYTE	7	6	5	4	3	2	1	0	BIT
0	块大小								Blcok Size - 块大小，不包含这个字节
1									Data Values - 块数据，8-bit的字符串
2									
...									
254									
255									

GIF结构

GIF文件的结构为：文件头（File Header）、GIF数据流（GIF Data Stream）和文件终结器（Trailer）三部分。

接下来详细讲解每部分的构成。

三.GIF各部分组成

1.文件头

文件头包含文件署名（Signature）和版本号（Version）两部分。

(1)文件署名

文件署名表示该文件是GIF文件，这部分由"GIF"三个字符组成：

BYTE	7	6	5	4	3	2	1	0	BIT
0	'G'(0x47)								文件署名
1	'I'(0x49)								
2	'F'(0x46)								

(2)版本号

版本号表示GIF版本号，有"87a"和"89a"两种：

BYTE	7	6	5	4	3	2	1	0	BIT
0	'8'(0x38)								版本号
1	'7'(0x39)或'9'(0x40)								
2	"a"(0x61)								

2.GIF数据流

GIF数据流包含逻辑屏幕标识符（Logical Screen Descriptor）、全局颜色列表（Global Color Table）和图像块（Image Block）三个基本部分以及一些扩展部分组成，扩展部分包括：图形控制扩展（Graphic Control Extension）、注释扩展（Comment Extension）、图形文本扩展（Plain Text Extension）以及应用程序扩展（Application Extension）四部分。

(1)逻辑屏幕标识符

该部分由7个字节组成，结构为：

BYTE	7	6	5	4	3	2	1	0	BIT
0	逻辑屏幕宽度								像素数，定义GIF图像宽度
1									
2	逻辑屏幕高度								像素数，定义GIF图像高度
3									
4	m	cr			s	pixel			具体描述见下
5	背景色								背景颜色（在全局颜色列表中的索引，如果没有全局颜色列表，则该值没有意义）
6	像素宽高比								像素宽高比

- m：全局颜色列表标志，当置位时表示有全局颜色列表，pixel值有意义

- cr: 颜色深度, cr+1确定图像的颜色深度
- s: 分类标志, 如果置位表示全局颜色列表分类排序
- pixel: 全局颜色列表大小, pixel+1确定颜色列表的索引数 (2的pixel+1次方)

(2)全局颜色列表

全局颜色列表需要紧跟在逻辑屏幕标识符后, 颜色列表中保存颜色的索引, 其中每个索引又由三个字节 (代表R、G、B) 组成:

BYTE	7	6	5	4	3	2	1	0
0	索引1的R值							
1	索引1的G值							
2	索引1的B值							
3	索引2的R值							
4	索引2的G值							
5	索引2的B值							
6	...							

(3)图像块

图像块由图像标识符 (Image Descriptor)、局部颜色列表 (Local Color Table) 和基于颜色列表的图像数据 (Table-Based Image Data) 组成。

①图像标识符

一个GIF中可以包含多张图片, 新的图像以图像标识符开始, 该部分由**10个字节**组成:

BYTE	7	6	5	4	3	2	1	0	BIT
0	0	0	1	0	1	1	0	0	图像标识符开始，固定值',' (0x2C)
1	X方向偏移量								必须限制在逻辑屏幕尺寸范围内
2									
3	Y方向偏移量								
4									
5	图像宽度								
6									
7	图像高度								
8									
10	m	i	s	r		pixel			具体描述见下

- m：局部颜色列表标志，当置位时表示图像标识符后有一个局部颜色列表，pixel值有意义
- i：交织标志，置位时图像数据使用交织方式排列，否则使用顺序排列
- s：分类标志，如果置位表示局部颜色列表分类排序
- r：保留，必须初始化为0
- pixel：局部颜色列表大小，pixel+1确定颜色列表的索引数（2的pixel+1次方）

②局部颜色列表

结构格式与全部颜色列表一致：

BYTE	7	6	5	4	3	2	1	0
0	索引1的R值							
1	索引1的G值							
2	索引1的B值							
3	索引2的R值							
4	索引2的G值							
5	索引2的B值							
6	...							

③基于颜色列表的图像数据

图像数据是由LZW编码长度和N个数据块组成的，解压后数据为该帧图片每个像素点颜色所对应的颜色索引：

LZW编码长度

BYTE	7	6	5	4	3	2	1	0	BIT
0	LZW编码长度								LZW编码初始码表大小的位数

N个数据块

连续的N个数据块，其中最后一个数据块是由长度为0的数据块组成，即：

BYTE	7	6	5	4	3	2	1	0	BIT
	...								N组数据图像
	数据块								
	...								
	0x00								数据块大小为0，作为一组数据块的收尾

(4)图形控制扩展

该部分需要89a版本，可以放在一个图像块或文本扩展块前面，用来控制后面的图像或文本的渲染形式，结构如下：

BYTE	7	6	5	4	3	2	1	0	BIT
0	扩展块标识								扩展块开始标识，固定值0x21
1	图形控制扩展标识								图形控制扩展标识，固定值0xF9
2	块大小								不包含块终结器，固定值0x04
3	保留			处置方式		i	t		具体描述如下
4	延迟时间								单位1/100秒，如果值不为1，表示暂停规定时间后再继续处理下面数据流
5									
6	透明色索引								透明色索引值
7	块终结器								标识块终结，固定值0x00

- 处置方法：指出处置图形的方法，当值为：0 - 不使用处置方法；1 - 不处置图形，把图形从当前位置移去；2 - 回复到背景色；3 - 回复到先前状态；4-7 - 自定义
- i：用户输入标志，指出是否期待用户有输入之后才继续进行下去，置位表示期待，值否表示不期待。用户输入可以是按回车键、鼠标点击等，可以和延迟时间一起使用，在设置的延迟时间内用户有输入则马上继续进行，或者没有输入直到延迟时间到达而继续
- t：透明颜色标志，置位表示使用透明颜色

(5)注释扩展

该部分需要89a版本，可以用来记录图形、版权、描述等任何的非图形和控制的纯文本数据(7-bit ASCII 字符)，注释扩展并不影响对图象数据流的处理，解码器完全可以忽略它。存放位置可以是数据流的任何地方，最好不要妨碍控制和数据块，推荐放在数据流的开始或结尾，结构如下：

BYTE	7	6	5	4	3	2	1	0	BIT
0	扩展块标识								扩展块开始标识，固定值0x21
1	注释扩展标识								注释控制扩展标识，固定值0xFE
	...								N组数据块
	注释块								
	...								
	0x00								数据块大小为0，作为一组数据块的收尾

(6)图形文本扩展

该部分需要89a版本，用来绘制一个简单的文本图象，这一部分由用来绘制的纯文本数据（7-bit ASCII 字符）和控制绘制的参数等组成。绘制文本借助于一个文本框（Text Grid）来定义边界，在文本框中划分多个单元格，每个字符占用一个单元，绘制时按从左到右、从上到下的顺序依次进行，直到最后一个字符或者占满整个文本框（之后的字符将被忽略，因此定义文本框的大小时应该注意到是否可以容纳整个文本），绘制文本的颜色索引使用全局颜色列表，没有则可以使用一个已经保存的前一个颜色列表。另外，图形文本扩展块也属于图形块(Graphic Rendering Block)，可以在它前面定义图形控制扩展对它的表现形式进一步修改，结构如下：

BYTE	7	6	5	4	3	2	1	0	BIT
0	扩展块标识								扩展块开始标识，固定值0x21
1	注释扩展标识								注释控制扩展标识，固定值0x01
2	块大小								块大小，固定值0x0C
3	文本框左边界位置								像素值，文本框离逻辑屏幕的左边界距离
4									
5	文本框上边界位置								像素值，文本框离逻辑屏幕的上边界距离
6									
7	文本框宽度								像素值
8									
9	文本框高度								像素值
10									
11	字符单元格宽度								像素值，单个单元格宽度
12	字符单元格高度								像素值，单个单元格高度
13	文本前景色索引								前景色在全局颜色列表中的索引
14	文本背景色索引								背景色在全局颜色列表中索引
									N组数据块

	文本数据块								

	0x00								

(7)应用程序扩展

该部分需要89a版本，应用程序可以在这里定义自己的标识、信息等，结构如下：

BYTE	7	6	5	4	3	2	1	0	BIT
0	扩展块标识								扩展块开始标识，固定值0x21
1	应用程序扩展标识								应用程序扩展标识，固定值0xFF
2	块大小								块大小，固定值0x0B
3	应用程序标识符								用来鉴别应用程序自身的标识（8个连续的ASCII字符）
4									
5									
6									
7									
8									
9									
10									
11	应用程序鉴别码								应用程序定义的特殊标识码（3个连续ASCII字符）
12									
13									
	...								N组数据块
	应用程序数据								
	...								
	0x00								数据块大小为0，作为一组数据块的收尾

其中，现在最多使用的是网景公司定义的应用程序控制标识，该标识中定义了GIF播放的循环次数，结构如下：

BYTE	7	6	5	4	3	2	1	0	BIT		
0	扩展块标识								扩展块开始标识, 固定值0x21		
1	应用程序扩展标识								应用程序扩展标识, 固定值0xFF		
2	块大小								块大小, 固定值0x0B		
3	0x4E('N')								NETSCAPE		
4	0x45('E')										
5	0x54('T')										
6	0x53('S')										
7	0x43('C')										
8	0x41('A')										
9	0x50('P')										
10	0x45('E')										
11	0x32('2')								2.0		
12	0x2E('.')										
13	0x30('0')										
14	0x03								该数据块长度为3字节(不包含该字节)	数据块	
15	0x01								数据块id		
16	循环次数								循环次数		
17											
18	0x00								数据块大小为0, 作为一组数据块的收尾		

3.文件终结器

这一部分只有一个标识符为0x3B, 标识一个GIF文件结束.

BYTE	7	6	5	4	3	2	1	0	BIT	
0	文件终结器								GIF Trailer - 标识GIF文件结束, 固定值0x3B	

以上就是GIF文件的结构, 下一篇文章将在iOS平台上对GIF文件直接操作进而达到对GIF调速的功能。

