游戏开发字符编码分析

2015-08-26

字符编码问题看似很小，不受到大家重视，但实际上很容易引发一些莫名其妙的问题。结合自己在开发过程中遇到的问题，总结一些普及性的知识，希望对大家有所帮助。

目录

1

必知基础

基本概念

位　　计算机处理信息的最小单位

字节　计算机处理信息的基本单位（计量单位）

编码　编码的过程是将字符转换成字节流

解码　解码的过程是将字节流解析为字符

字符（Character）

字符是文字与符号的总称

字符集（Character Set）

指的是一组抽象的已编号的字符的有序集合（不一定是连续）

字符码（Code Point）

指的就是字符集中每个字符的数字编号

字符编码（Character Encoding）

计算机要处理各种字符，就需要将字符和二进制内码对应起来，这种对应关系就是字符编码

字符集和字符编码

字符集（字符集方案）

ASCII
GB 2312
Big5
GBK
GB 18030
UNICODE

字符编码（编码方案）

ASCII
GB 2312
Big5
GBK
GB 18030
UTF-8
UTF-16
UTF-32

代码页（Code Page）

微软为了适应世界上不同地区用户的文化背景和生活习惯,在Windows中设计了区域(Locale)设置的功能。

Windows 里说的「ANSI」其实是 Windows code pages，这个模式根据当前 locale 选定具体的编码，比如简中 locale 下是 GBK。

第 2 章

常用字符集和字符编码

ASCII

ASCII（American Standard Code for Information Interchange，美国讯息交换标准代码）是一套基于英文字母的**单字节**字符编码系统，最初是美国国家标准协会 ASA（今日的美国国家标准协会，**ANSI** ），供不同计算机在相互通信时用作共同遵守的西文字符编码标准，它已被国际标准化组织（International Organization for Standardization, **ISO**）定为国际标准，称为**ISO 646**标准。

适用于所有拉丁文字字母。

ASCII码是上个世纪最流行的编码体系之一。

ASCII码表

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

ASCII扩展码

ASCII 字符代码表 二

高四位 / 低四位		扩充ASCII码字符集															
		1000		1001		1010		1011		1100		1101		1110		1111	
		8		9		A/10		B/16		C/32		D/48		E/64		F/80	
		+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符	+进制	字符
0000	0	128	Ç	144	É	160	á	176	▒	192	└	208	╨	224	α	240	≡
0001	1	129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±
0010	2	130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
0011	3	131	â	147	ô	163	ú	179	│	195	├	211	╙	227	π	243	≤
0100	4	132	ä	148	ö	164	ñ	180	┤	196	─	212	Ô	228	Σ	244	⌠
0101	5	133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
0110	6	134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	μ	246	÷
0111	7	135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
1000	8	136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
1001	9	137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	•
1010	A	138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
1011	B	139	ï	155	¢	171	½	187	╗	203	╦	219	▉	235	δ	251	√
1100	C	140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n
1101	D	141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
1110	E	142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
1111	F	143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	BLANK FF

注：表中的ASCII字符可以用：ALT + "小键盘上的数字键" 输入

ASCII码大致可以分作三部分組成。
第一部分是：ASCII非打印控制字符；
第二部分是：ASCII打印字符；
第三部分是：扩展ASCII打印字符

由80H到0FFH共128个字符是有IBM制定的**非标准**ASCII码，这些字符用来表示框线，音标和其他欧洲非英语系的字母。

汉字系列

国家标准强制标准冠以“GB”。

GB系列常用的有：
GB2312
GBK
GB18030

GB-2312

GB-2312 是一个**区位码**形式的字符集标准，不过实际上基本都用 EUC-CN 来编码，兼容于ASCII。

GB-2312标准共收录6763个汉字，其中一级汉字3755个，二级汉字3008个；同时，GB 2312收录了包括拉丁字母、希腊字母、日文平假名及片假名字母、俄语西里尔字母在内的682个全角字符。

GB-2312的出现，基本满足了汉字的计算机处理需要，它所收录的汉字已经覆盖中国大陆99.75%的使用频率。

GB-2312中对所收汉字进行了“分区”处理，每区含有94(16*6-2)个汉字/符号。这种表示方式也称为区位码。
01-09区(高位0xA1-0xA7，低位0xA1-0xFE)为特殊符号。
16-55区(高位0xB0-0xD7，低位0xA1-0xFE)为一级汉字，按拼音排序。
56-87区(高位0xD0-0xF7，低位0xA1-0xFE)为二级汉字，按部首/笔画排序。
10-15区及88-94区则未有编码。

举例来说，“啊”字是GB2312之中的第一个汉字，它的区位码就是1601,对应字符码为0xB0A1(高位16+0xA0＝0xB0，低位01+0xA0 = 0xA1))。

GBK

1995年12月发布的汉字编码国家标准，是对GB2312编码的扩充，对汉字采用双字节编码。GBK字符集共收录21003个汉字，包含国家标准GB13000-1中的全部中日韩汉字，和**BIG5**编码（台湾地区繁体中文标准字符集，采用双字节编码，共收录13053个中文字，1984年实施）中的所有汉字。

GB18030

2000年3月17日发布的汉字编码国家标准，是对GBK编码的扩充，覆盖中文、日文、朝鲜语和中国少数民族文字，其中收录27484个汉字。GB18030字符集采用单字节、双字节和四字节三种方式对字符编码。兼容GBK和GB2312字符集。

SBCS、DBCS和MBCS

SBCS、DBCS和MBCS分别是单字节字符集、双字节字符集和多字节字符集的缩写。

SBCS、DBCS和MBCS的最大编码长度分别是1字节、两字节和大于两字节（例如4或5字节）。

例如：

代码页1252 （ANSI-拉丁文 I）是单字节字符集；

代码页936 （ANSI/OEM-简体中文 GBK）是双字节字符集；

代码页54936 （GB18030 简体中文）是多字节字符集。

UNICODE字符集

Unicode 是为了解决传统的字符编码方案的局限而产生的，它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。

Unicode就像一个电话本，标记着字符和数字之间的映射关系。因为它们可能是随机指定的，而不会给出任何解释。总是用U+开头。理论上每种语言中的每种字符都在Unicode字符集中有一个对应的字符码。

UTF-8

UTF-8（8-bit Unicode Transformation Format）是一种针对Unicode的可变长度字符编码，也是一种前缀码。

它可以用来表示Unicode标准中的任何字符。

编码规则：

对于某一个字符的UTF-8编码，如果只有一个字节则其最高二进制位为0；如果是多字节，其第一个字节从最高位开始，连续的二进制位值为1的个数决定了其编码的位数，其余各字节均以10开头。UTF-8最多可用到6个字节。

Unicode编码(十六进制)	UTF-8 字节流(二进制)
00000000 - 0000007F	0xxxxxxx
00000080 - 000007FF	110xxxxx 10xxxxxx
00000800 - 0000FFFF	1110xxxx 10xxxxxx 10xxxxxx
00010000 - 001FFFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
00200000 - 03FFFFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
04000000 - 7FFFFFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

UTF-16

UTF-16编码以16位无符号整数为单位，也是一种针对Unicode的的可变长度编码方案。
它使用2个或者4个字节来存储字符。

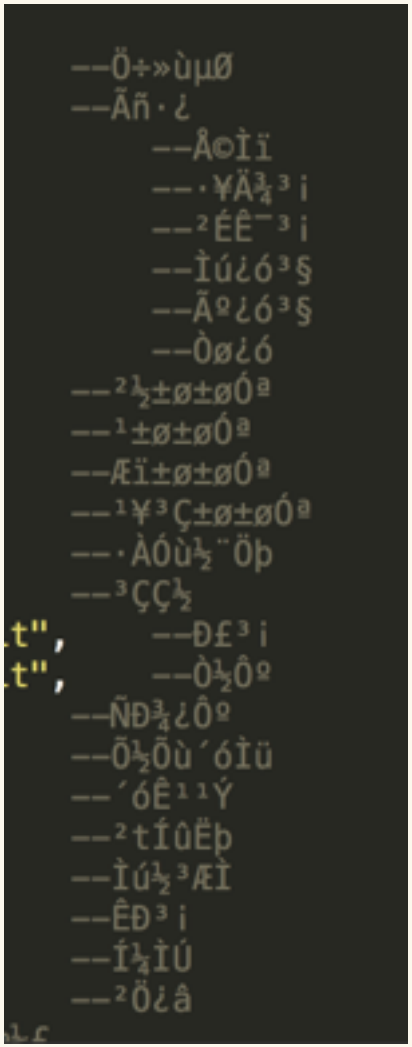
UCS-2

UCS-2 (2-byte Universal Character Set)是一种定长的编码方式，UCS-2仅仅简单的使用一个16位码元来表示码位，也就是说在0到0xFFFF的码位范围内，它和UTF-16基本一致。

第3章

应用

乱码



AppDelegate.cpp

AppDelegate.cpp istCharacterEncoding 〉 Classes 〉 AppDelegate.cpp 〉 AppDelegate::applicationDidFinishLaunching()

```
51      return 0; //flag for packages manager
52  }
53
54  //进入游戏
55  bool AppDelegate::applicationDidFinishLaunching()
56  {
57      // set default FPS
58      Director::getInstance()->setAnimationInterval(1.0 / 60.0f);
```

AppDelegate.cpp

AppDelegate | applicationDidFinishLaunching()

```
49  static int register_all_packages()
50  {
51      return 0; //flag for packages manager
52  }
53
54  //枵涘顑娓告垙
55  bool AppDelegate::applicationDidFinishLaunching()
56  {
57      // set default FPS
58      Director::getInstance()->setAnimationInterval(1.0 / 60.0f);
```

绵
C3E0
1100 0011 1110 0000　　gb2312 编码表
细
CFB8
1100 1111 1001 1000

奇怪的“联通”
C1 AA CD A8
1100 0001 1010 1010 1100 1101 1010 1000

BOM

再见编码问题 unexpected symbol near '\357'stack traceback:

16进制编辑器

用UTF-8编码格式执行rb文件时，给出如下提示：
Invalid char `\357' in expression
Invalid char `\273' in expression
Invalid char `\277' in expression

原因：由TF-8的最前面有EF BB BF这三个隐藏的字符导致的以上错误。

解决方案：用16进制编辑器打开该文件，将EF BB BF这三个char去掉即可。（用UltraEdit或其他带有16进制编辑器）

编译器处理问题

//注释
do something

/** **/

限制编码范围

输入框的限制

CCFilter

编码处理

很多时候我们引擎提供的功能不足满足我们的需求，需要我们自行补充。

例子：

输入框字符串的删除处理

字符串处理

注意一个问题

字符串数组末尾注意加'\0'

dragon 〉 Classes 〉 frUtil 〉 FRMD5.cpp 〉 Decode(const std::string& szToDecode)

```
std::vector<unsigned char> Decode( const std::string& szToDecode )
{
    std::vector<unsigned char> result;

    int hex = 0;

    for ( size_t i = 0; i < szToDecode.length(); ++i )
    {...}
    //字符串末尾要加个结束符号 否则会导致后面会出现不同平台处理不同的转换问题 mixi 15.03.10
    result.push_back('\0');
    return result;
}
```

内码支持问题

字体显示过程

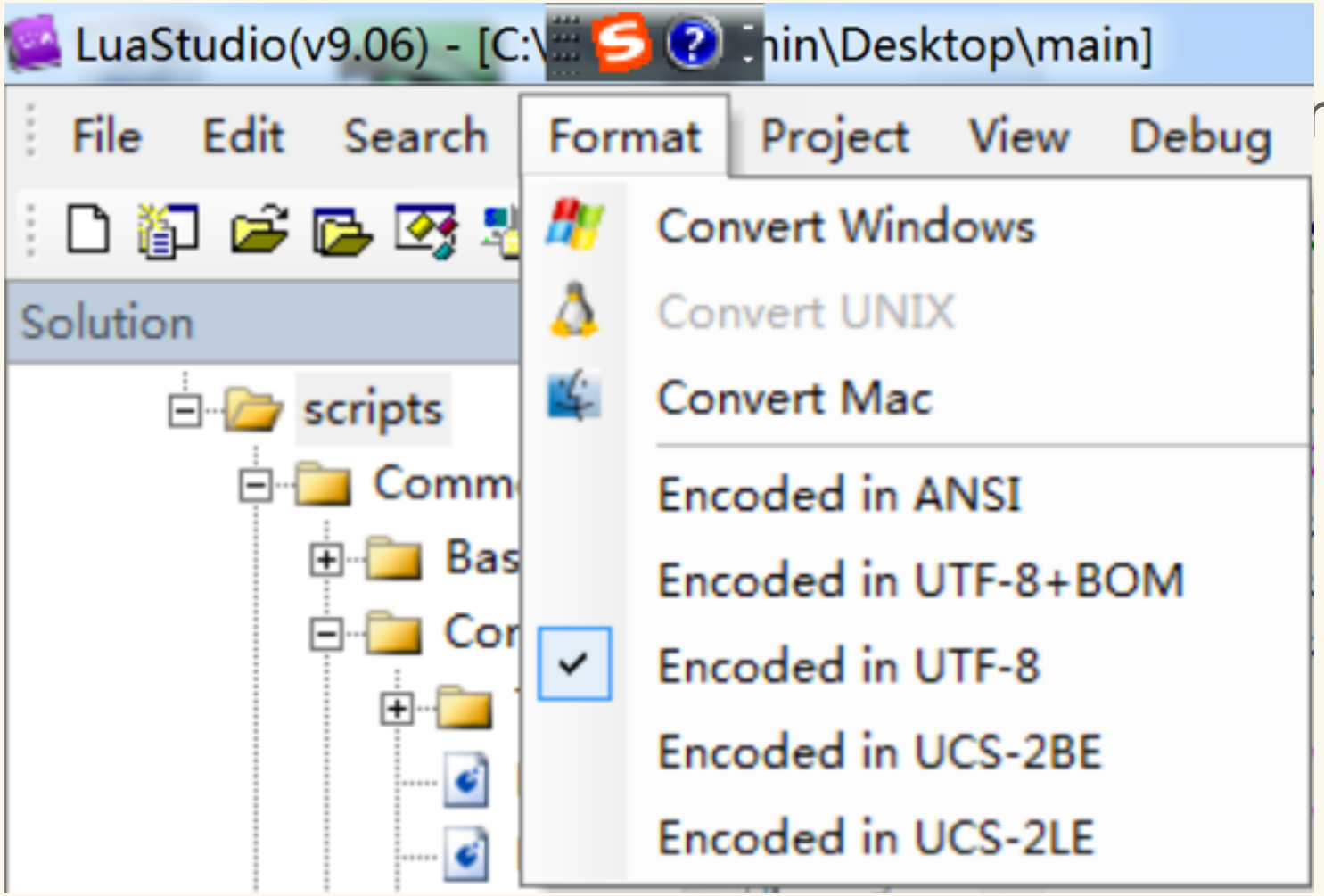
字符码（Unicode）
→字体的字形索引（Glyph ID）
→字形轮廓
→点阵图字形

艺术字

做游戏时候有时会将数字做成艺术字，美术人员预先做出一些字符的图片。
通常会是 012345679，这个顺序是按照ASCII排列的。

```
TextAtlas* TextAtlas::create(const std::string &stringValue,
                             const std::string &charMapFile,
                             int itemWidth,
                             int itemHeight,
                             const std::string &startCharMap)
```

应用软件中的对应关系



mation

一句话建议：涉及兼容性考量时，不要用记事本，用专业的文本编辑器保存为不带 BOM 的 UTF-8。

一句话提醒：要是有在xcode上正常，vs上有问题的，请考虑下中文注释问题。