

Engr 123

Notes on Console Graphics in C#

The term "console graphics" refers to drawing on the console screen using cursor manipulation and writing ASCII characters instead of setting individual pixels. Console graphics is a somewhat crude way to do graphics but it is adequate for some simple games and such.

ASCII Codes

In console graphics the smallest unit we can put on the screen is a character. Since characters cannot be transmitted over wires, a bit-code was developed for use in telegraphy which each character was represented by a bit pattern. As early as 1874 Emile Baudot patented the Baudot code for use with the telegraph. It used five-bits for each letter allowing representation of up to $2^5 = 32$ characters. In the 1960s this developed into a 7-bit American Standard Code for Information Interchange (ASCII). With 7-bits up to 128 characters could be represented. Paper-tape with punched holes, in use at the time, could handle 8-bits and the 8th bit was used as a parity bit to detect an odd-number of bit-errors in transmission. IBM, which had more than 80% of the computer market had its own encoding scheme called Extended Binary Coded Decimal Interchange Code (EBCDIC) which was an 8-bit code.

With the introduction of the IBM PC and the DOS operating system in 1980, it became common practice to use the 8th bit for additional character encoding instead of a parity bit for error correction. DOS, Windows, and the McIntosh all had their own encoding schemes which used the 8th bit to extend the basic ASCII code. The Windows encoding scheme is called Windows-1252 and is similar (but not identical to) the international standard ISO 8859-15. The table on the following pages shows the encoding scheme for Windows-1252. There are two pages with three columns per page. Each column gives the 8-bit character code in hexadecimal, the printed character, the 16-bit Unicode in hexadecimal, and an abbreviated description.

To get the actual binary encoding you need to change the hexadecimal code to binary using the following table:

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

For example the letter capital A is listed as 41 in hexadecimal which corresponds to 0100 0001 in binary

Code	Char	Unicode	Description	32	2	0032	DIGIT TWO	65	e	0065	SMALL E
00		0000	NULL	33	3	0033	DIGIT THREE	66	f	0066	SMALL F
01		0001	START OF HEADING	34	4	0034	DIGIT FOUR	67	g	0067	SMALL G
02		0002	START OF TEXT	35	5	0035	DIGIT FIVE	68	h	0068	SMALL H
03		0003	END OF TEXT	36	6	0036	DIGIT SIX	69	i	0069	SMALL I
04		0004	END OF TRANSMISSION	37	7	0037	DIGIT SEVEN	6A	j	006A	SMALL J
05		0005	ENQUIRY	38	8	0038	DIGIT EIGHT	6B	k	006B	SMALL K
06		0006	ACKNOWLEDGE	39	9	0039	DIGIT NINE	6C	l	006C	SMALL L
07		0007	BELL	3A	:	003A	COLON	6D	m	006D	SMALL M
08		0008	BACKSPACE	3B	;	003B	SEMICOLON	6E	n	006E	SMALL N
09		0009	HORIZONTAL TABULATION	3C	<	003C	LESS-THAN SIGN	6F	o	006F	SMALL O
0A		000A	LINE FEED	3D	=	003D	EQUALS SIGN	70	p	0070	SMALL P
0B		000B	VERTICAL TABULATION	3E	>	003E	GREATER-THAN SIGN	71	q	0071	SMALL Q
0C		000C	FORM FEED	3F	?	003F	QUESTION MARK	72	r	0072	SMALL R
0D		000D	CARRIAGE RETURN	40	@	0040	COMMERCIAL AT	73	s	0073	SMALL S
0E		000E	SHIFT OUT	41	A	0041	CAPITAL A	74	t	0074	SMALL T
0F		000F	SHIFT IN	42	B	0042	CAPITAL B	75	u	0075	SMALL U
10		0010	DATA LINK ESCAPE	43	C	0043	CAPITAL C	76	v	0076	SMALL V
11		0011	DEVICE CONTROL ONE	44	D	0044	CAPITAL D	77	w	0077	SMALL W
12		0012	DEVICE CONTROL TWO	45	E	0045	CAPITAL E	78	x	0078	SMALL X
13		0013	DEVICE CONTROL THREE	46	F	0046	CAPITAL F	79	y	0079	SMALL Y
14		0014	DEVICE CONTROL FOUR	47	G	0047	CAPITAL G	7A	z	007A	SMALL Z
15		0015	NEGATIVE ACKNOWLEDGE	48	H	0048	CAPITAL H	7B	{	007B	LEFT CURLY BRACKET
16		0016	SYNCHRONOUS IDLE	49	I	0049	CAPITAL I	7C		007C	VERTICAL LINE
17		0017	END OF TRANS BLOCK	4A	J	004A	CAPITAL J	7D	}	007D	RIGHT CURLY BRACKET
18		0018	CANCEL	4B	K	004B	CAPITAL K	7E	~	007E	TILDE
19		0019	END OF MEDIUM	4C	L	004C	CAPITAL L	7F	☒	007F	DELETE
1A		001A	SUBSTITUTE	4D	M	004D	CAPITAL M	80	€	20AC	EURO SIGN
1B		001B	ESCAPE	4E	N	004E	CAPITAL N	82	,	201A	SINGLE QUOTATION MARK
1C		001C	FILE SEPARATOR	4F	O	004F	CAPITAL O	83	f	0192	SMALL F WITH HOOK
1D		001D	GROUP SEPARATOR	50	P	0050	CAPITAL P	84	„	201E	DOUBLE QUOTATION MARK
1E	-	001E	RECORD SEPARATOR	51	Q	0051	CAPITAL Q	85	...	2026	HORIZONTAL ELLIPSIS
1F		001F	UNIT SEPARATOR	52	R	0052	CAPITAL R	86	†	2020	DAGGER
20		0020	SPACE	53	S	0053	CAPITAL S	87	‡	2021	DOUBLE DAGGER
21	!	0021	EXCLAMATION MARK	54	T	0054	CAPITAL T	88	^	02C6	MODIFIER ACCENT
22	"	0022	QUOTATION MARK	55	U	0055	CAPITAL U	89	‰	2030	PER MILLE SIGN
23	#	0023	NUMBER SIGN	56	V	0056	CAPITAL V	8A	Š	0160	CAPITAL S WITH CARON
24	\$	0024	DOLLAR SIGN	57	W	0057	CAPITAL W	8B	<	2039	ANGLE QUOTATION MARK
25	%	0025	PERCENT SIGN	58	X	0058	CAPITAL X	8C	Œ	0152	CAPITAL LIGATURE OE
26	&	0026	AMPERSAND	59	Y	0059	CAPITAL Y	8E	Ž	017D	CAPITAL Z WITH CARON
27	'	0027	APOSTROPHE	5A	Z	005A	CAPITAL Z	91	‘	2018	LEFT QUOTATION MARK
28	(0028	LEFT PARENTHESIS	5B	[005B	LEFT SQUARE BRACKET	92	’	2019	RIGHT QUOTATION MARK
29)	0029	RIGHT PARENTHESIS	5C	\	005C	REVERSE SOLIDUS	93	“	201C	LEFT QUOTATION MARK
2A	*	002A	ASTERISK	5D]	005D	RIGHT SQUARE BRACKET	94	”	201D	RIGHT QUOTATION MARK
2B	+	002B	PLUS SIGN	5E	^	005E	CIRCUMFLEX ACCENT	95	•	2022	BULLET
2C	,	002C	COMMA	5F	~	005F	LOW LINE	96	–	2013	EN DASH
2D	-	002D	HYPHEN-MINUS	60	`	0060	GRAVE ACCENT	97	—	2014	EM DASH
2E	.	002E	FULL STOP	61	a	0061	SMALL A	98	~	02DC	SMALL TILDE
2F	/	002F	SOLIDUS	62	b	0062	SMALL B	99	™	2122	TRADE MARK SIGN
30	0	0030	DIGIT ZERO	63	c	0063	SMALL C	9A	š	0161	SMALL S WITH CARON
31	1	0031	DIGIT ONE	64	d	0064	SMALL D	9B	>	203A	ANGLE QUOTATION MARK

9C	œ	0153	SMALL LIGATURE OE	BE	¾	00BE	FRACTION THREE QUARTERS	DF	ß	00DF	SMALL SHARP S
9E	ž	017E	SMALL Z WITH CARON	BF	¿	00BF	INVERTED QUESTION MARK	E0	à	00E0	SMALL A WITH GRAVE
9F	ÿ	0178	Y WITH DIAERESIS	C0	À	00C0	CAPITAL A WITH GRAVE	E1	á	00E1	SMALL A WITH ACUTE
A0		00A0	NO-BREAK SPACE	C1	Á	00C1	CAPITAL A WITH ACUTE	E2	â	00E2	SMALL A CIRCUMFLEX
A1	¡	00A1	EXCLAMATION MARK	C2	Â	00C2	CAPITAL A CIRCUMFLEX	E3	ã	00E3	SMALL A WITH TILDE
A2	¢	00A2	CENT SIGN	C3	Ã	00C3	CAPITAL A WITH TILDE	E4	ä	00E4	SMALL A WITH DIAERESIS
A3	£	00A3	POUND SIGN	C4	Ä	00C4	CAPITAL A DIAERESIS	E5	å	00E5	SMALL A RING ABOVE
A4	¤	00A4	CURRENCY SIGN	C5	Å	00C5	CAPITAL A RING ABOVE	E6	æ	00E6	SMALL AE
A5	¥	00A5	YEN SIGN	C6	Æ	00C6	CAPITAL AE	E7	ç	00E7	SMALL C WITH CEDILLA
A6	¦	00A6	BROKEN BAR	C7	Ç	00C7	CAPITAL C WITH CEDILLA	E8	è	00E8	SMALL E WITH GRAVE
A7	§	00A7	SECTION SIGN	C8	È	00C8	CAPITAL E WITH GRAVE	E9	é	00E9	SMALL E WITH ACUTE
A8	¨	00A8	DIAERESIS	C9	É	00C9	CAPITAL E WITH ACUTE	EA	ê	00EA	SMALL E CIRCUMFLEX
A9	©	00A9	COPYRIGHT SIGN	CA	Ê	00CA	CAPITAL E CIRCUMFLEX	EB	ë	00EB	SMALL E WITH DIAERESIS
AA	ª	00AA	FEMININE ORD INDICATOR	CB	Ë	00CB	CAPITAL E WITH GRAVE	EC	ì	00EC	SMALL I WITH GRAVE
AB	«	00AB	ANGLE QUOTATION MARK	CC	Ì	00CC	CAPITAL I WITH GRAVE	ED	í	00ED	SMALL I WITH ACUTE
AC	¬	00AC	NOT SIGN	CD	Í	00CD	CAPITAL I WITH ACUTE	EE	î	00EE	SMALL I CIRCUMFLEX
AD		00AD	SOFT HYPHEN	CE	Î	00CE	CAPITAL I CIRCUMFLEX	EF	ï	00EF	SMALL I WITH DIAERESIS
AE	®	00AE	REGISTERED SIGN	CF	Ï	00CF	CAPITAL I DIAERESIS	F0	ð	00F0	SMALL ETH
AF	ˉ	00AF	MACRON	D0	Ð	00D0	CAPITAL ETH	F1	ñ	00F1	SMALL N WITH TILDE
B0	°	00B0	DEGREE SIGN	D1	Ñ	00D1	CAPITAL N WITH TILDE	F2	ò	00F2	SMALL O WITH GRAVE
B1	±	00B1	PLUS-MINUS SIGN	D2	Ò	00D2	CAPITAL O WITH GRAVE	F3	ó	00F3	SMALL O WITH ACUTE
B2	²	00B2	SUPERSCRIP TWO	D3	Ó	00D3	CAPITAL O WITH ACUTE	F4	ô	00F4	SMALL O CIRCUMFLEX
B3	³	00B3	SUPERSCRIP THREE	D4	Ô	00D4	CAPITAL O CIRCUMFLEX	F5	õ	00F5	SMALL O WITH TILDE
B4	´	00B4	ACUTE ACCENT	D5	Õ	00D5	CAPITAL O WITH TILDE	F6	ö	00F6	SMALL O WITH DIAERESIS
B5	µ	00B5	MICRO SIGN	D6	Ö	00D6	CAPITAL O DIAERESIS	F7	÷	00F7	DIVISION SIGN
B6	¶	00B6	PILCROW SIGN	D7	×	00D7	MULTIPLICATION SIGN	F8	ø	00F8	SMALL O WITH STROKE
B7	·	00B7	MIDDLE DOT	D8	∅	00D8	CAPITAL O WITH STROKE	F9	ù	00F9	SMALL U WITH GRAVE
B8	,	00B8	CEDILLA	D9	Ù	00D9	CAPITAL U WITH GRAVE	FA	ú	00FA	SMALL U WITH ACUTE
B9	¹	00B9	SUPERSCRIP ONE	DA	Ú	00DA	CAPITAL U WITH ACUTE	FB	û	00FB	SMALL U CIRCUMFLEX
BA	º	00BA	MASCULINE ORD INDICATOR	DB	Û	00DB	CAPITAL U CIRCUMFLEX	FC	ü	00FC	SMALL U WITH DIAERESIS
BB	»	00BB	ANGLE QUOTATION MARK	DC	Ü	00DC	CAPITAL U DIAERESIS	FD	ý	00FD	SMALL Y WITH ACUTE
BC	¼	00BC	FRACTION ONE QUARTER	DD	Ý	00DD	CAPITAL Y WITH ACUTE	FE	þ	00FE	SMALL THORN
BD	½	00BD	VULGAR FRACTION ONE HALF	DE	ƒ	00DE	CAPITAL THORN	FF	ÿ	00FF	SMALL Y WITH DIAERESIS

Console Operations

When you write a console application in C# the window defaults to a size of 30 rows (0 to 29) and 120 columns (0 to 119). This may not be true on all systems. You can see your console window size using the following operations in a console application.

```
Console.WriteLine(Console.WindowLeft);
Console.WriteLine(Console.WindowWidth);
Console.WriteLine(Console.WindowTop);
Console.WriteLine(Console.WindowHeight);
```

If you are using console graphics you should also set the encoding scheme for your console window:

```
Console.OutputEncoding = System.Text.Encoding.GetEncoding(1252);
```

If you want you can set the window title as well:

```
Console.Title = "Assignment 4 Gravity";
```

Other useful operations:

To clear the screen:

```
Console.Clear();
```

To set the cursor position:

```
Console.SetCursorPosition(column, row);
```

Note that column and row are both integers and must be within range of the current screen.

To write a character to the current cursor position:

```
Console.Write((char) 65);
Console.Write((char) 0x41);
Console.Write('A');
```

All three of these write the letter capital A to the screen since $65_{10} = 41_{16} = \text{ASCII code for 'A'}$. Also, the cursor position advances one place to the right each time you write a character. You can also use `Console.WriteLine` to get a new line after writing. To erase a character you can write a space (0x20) over the character to be erased.

To change the foreground color or the background color at the current position:

```
Console.ForegroundColor = ConsoleColor.Blue;
Console.BackgroundColor = ConsoleColor.White;
```

To change the console window size:

```
Console.BufferHeight = 60;
Console.WindowHeight = 52;
Console.BufferWidth = 200;
Console.WindowWidth = 180;
```

The console window has both a size (in columns and rows) and a buffer for storing characters. The buffer height must be greater than or equal to the window height and the buffer width must be greater than or equal to the window width. In addition the window height and width must be such that it can fit on the monitor at the resolution being used – this differs from system to system. If the buffer height is larger than the window height a scroll bar appears. Likewise for the buffer width. For general applications, it is best not to alter these values and to use the window height and width that is defined when your program starts.

To wait for the user to push a key:

```
Console.ReadKey(true);
```

This line returns when the user pushes any key. The argument is set to true so that the key is not displayed. If the argument is blank or false the key is displayed.