Hex2Sprite CSE Edition V 1.0

By TI-Freakware

<http://tifreakware.net>

[tifreak8x@hotmail.com](mailto:tifreak8x@hotmail.com)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Description
2. Using this program
3. Credits

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Description:

Hex2Sprite CSE is a hex to sprite displayer for the TI-84 Plus C Silver Edition calculators.

1. Using this program:

To use this program, you will need the following inputs:

Str9 - This is your hex string. Look below for the color usage definitions

|LSPR - This is the list variable that tells the program the following: {Sprite width, Row Start, Column Start, Display mode

The program uses/destroys A, B, C, D, E, F, and G.

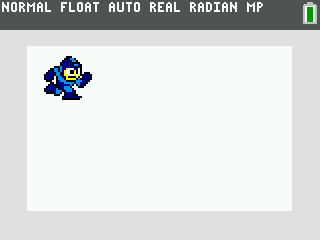
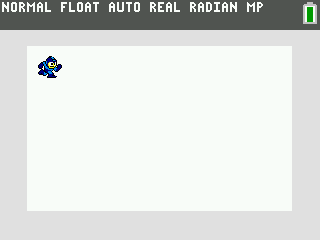
Str9 syntax: With the way this is set up, you can technically have any sprite size you want. The only downfall is the larger the sprite, the longer it takes to display. You can use the normal hex values of 0-F to get your colors, according to the following chart:

|  |  |
| --- | --- |
| 0 | Skip |
| 1 | Blue |
| 2 | Red |
| 3 | Black |
| 4 | Magenta |
| 5 | Green |
| 6 | Orange |
| 7 | Brown |
| 8 | Navy Blue |
| 9 | Light Blue |
| A | Yellow |
| B | White |
| C | Light Grey |
| D | Medium Grey |
| E | Grey |
| F | Dark Grey |
| G | Drop to next line, restart row |

As you’ll notice, I use hex character ‘G’ to help save space for hex values, to keep a person from needing a dozen 0s at the end of a line.

|LSPR is fairly self-explanatory.

1. Is defined as the sprite’s width, so if you define this as your max width, you won’t need a ‘G’ at the end of that particular line.
2. Is defined as the row you want to start displaying images to.
3. Is defined as the column you want to start displaying images to.
4. Is the display mode. Mode 1 displays single pixels at a time, while Mode 2 displays it in a 4:1 ratio (4 pixels for the normally 1 pixel).



The one on the left is displayed in Mode 1, while the one on the right is displayed in Mode 2.

000000000000333G

0000000000333993G

00000000038883993G

000000003888883333G

0000000038888839983G

0000033398888883383G

00033993988ABBB88B3G

0038399398ABB33A3B3G

0388839938ABB33A3B30333G

3888839938AABBBABA338883G

38883039938A3333A3038883G

38883039993AAAAA39338883G

03330038999333339998883G

0003303888999993998883G

003883388888893039883G

03888833889993000333G

388888933899993G

3883889993399983G

0333389930388883G

00000333003888333G

0000000003888888883G

0000000003333333333

Is an example of the sprite data used in the above images.

Once you have your sprite data saved into Str9, and your list created and stored to |LSPR, run prgmZSPR to see your creation appear on the graph screen. Please note that wherever you use a 0, it will not overwrite the sprite behind it, so if you are using a background color, that won’t get changed.

1. Credit

Credit goes to tifreak8x of TI-Freakware for building this program, such as it is. It is a hope that it will help spark an asm version in the very near future. If you need assistance using this program, feel free to join the forums at <http://tifreakware.net/forums/> or send me an email at [tifreak8x@hotmail.com](mailto:tifreak8x@hotmail.com) . Thanks for looking at my program!