VUCanvas - Version 1.1 - last revised 08/26/2000

If you want to contact me, my name is Benjamin J. Schneider (Ben) and my email is schneidb@ohio.edu. Feel free to drop me a post to let me know if you've improved on the class so I can incorporate your changes into the class that will be posted on my website. Also, let me know of any ideas you may have that should be incorporated that you may not know how to do. Whatever, feel free to contact me period. ;)

I would like to offer special thanks to Arthur Elsenaar. He is who added most of the PeakHold code. I think you'll find it works quite well.

VUCanvas - What is it?

This class is a subset of the canvas class and is a simple VUMeter for audio related projects. The VUCanvas can be used as a horizontal or a vertical VUMeter.

NOTE: Nothing is commented. This ReadMe file is the documentation. I know that many of you veteran programmers will find this irresponsible. I prefer conceptual documentation over "I did this to accomplish this..."

Instructions:

Note: Adjust the canvas to 128x7 for a horizontal VUMeter and 7x128 for a vertical VUMeter. Actually, 7 looks best to my eye, but the class will scale to different (widths in vertical orientation, heights in horizontal orientation) if you prefer. As of now, however, the class won't scale up or down so you're stuck with 128 high (or wide for horizontal orientation).

(Sub) - VUScale(Level as Integer) - Pass a level from 0 to 255 (the range that most Apple audio based calls uses) to have the VUCanvas show that level. This basically works like progressbar.value=x. It just looks better to use something that looks like a real VUMeter when dealing with audio applications.

(Property) - peakHold as Boolean - Use this to turn the Peak Hold option on or off Set this property to true to turn it on and set this property to false to turn it off.

(Property) - peakDuration as Integer - Use this to set the duration in 60th's of a second of the peakhold option. For example, if you want a peak hold of 2 seconds, you'd set this property to 120.

Note: There are other properties and methods in the class, but they are private because you don't need to access them to gain the full functionality of the class. You are welcome to look through them to see how they work as nothing is protected.

Enjoy,

Ben