Maximizing SPU RAM

V 1.0

Rob Vawter SCEA
2/13/98

A. Reducing the size of VAG data

Note:Not all of these methods may be used in tandem.

1. Removal of the SPU IRQ Clear Block

The last 16 bytes of each non-looping VAG are called an SPU IRQ Clear Block. These blocks ensure that an IRQ placed in a VAG body does not get spuriously called over and over. However, if your programmers are not using any of the libspu functions with "IRQ" in the name AND not using any of the SpuSt... VAG streaming functions, this block can be stripped off.

2. Setting all AIFF sizes to 28 sample boundaries

Everyone should know that AIFFs which will loop in VAG format must have loop markers set on 28 byte boundaries (both loop start and loop end, which is always sample end). This need not be the case for non-looping VAGs. But, any non-looping AIFF which has a remainder of 1 or 2 samples will be padded out with zero data during conversion to create what is basically a 28 sample divisible AIFF file anyway. If the sound artist can pare down some of these AIFF files, quite a few samples can be saved.

3. Downsampling

Downsampling allows for smaller sample data. AIFF2VAG supports from 44.1KHz all the way down to 5KHz.

4. SPU streaming

Allows for playback of VAGs larger than the available memory in SPU RAM. See libspu overview for details.

A. Increasing the amount of usable SPU RAM

1. Use of SPU RAM addresses 0x1000-0x100f

Hopefully, this feature will be available in library v. 4.3. This 16 byte block of memory contains another SPU IRQ Clear Block and can be overwritten under certain circumstances. See #A1 above for more details. Currently, however, SpuSetTransferStartAddr() (also called from within the libsnd transfer functions) will not allow 0x1000 as a valid address.

2. Create and use a "best fit" SPU RAM memory manager

SpuMalloc() is a "first fit" memory manager, not a "best fit" memory manager. A "first fit" memory manager walks the memory management list until it finds a block large enough to fit the requested size; that block is then allocated. A "best fit" memory manager walks the entire memory management list and chooses the block which is the smallest block greater than or equal to the requested size; that block is then allocated. A "best fit" memory manager will tend to fragment memory less, and therefore allow more sample data to be resident in SPU RAM for sound engines using dynamic loading. I have suggested code changes to make SpuMalloc() (also used by SsVabOpenHead) a "best fit" memory manager. Hopefully, this will be available by library version 4.3

C. Make the most of the available samples

1. Change sample pitch

Excited crowd noise VAG is the same as background crowd noise, only with a faster playback rate.

2. Using dynamic looping to lengthen sound effects

Hopefully, in an upcoming version of the libraries, functions will exist which allow dynamic looping of sounds to create the appearance of more sounds. For example, you could loop in the middle of a sound, lengthening the sound before, such as "Gooooooooooooooooooooooooooooooooal!" Therefore, the play-by-play announcer can lengthen or shorten his call dependent on the team, player, and situation.

3. Pitch LFO

Pitch LFO allows the playback rate of one muted voice to affect the playback of another. This can create different effects for the same sample data. See libspu overview for more details.

4. Noise Generator

Also known as a white noise generator. Could be used to create effects like TV static or a Geiger counter. See libspu overview for details

-