BYBYMONO demo readme

This is just a small readme to explain the different files in this small set of demo screens I made to say a nice goodbye to the monochrome part of the GFA BASIC video series 3.

Except for the music routine, the demo screens were completely written in pure GFA BASIC. So no extra libraries, just default commands. The screens rely mostly on using default drawing and text commands to create the graphics, BMOVE and RC_COPY to put the graphics in their memory locations, and then use BMOVE and LONG{} to create the animations.

DEMO versions

There are 2 versions of this demo. The code in both is mostly the same. The EMU_60HZ version is meant for running on a emulator that you can sync to 60Hz for smooth animation and Capturing. Since vsync to 60Hz makes the emulated machine run slower I play the music at 60Hz instead of the default 50Hz, and I also adjusted the pause to make the sprite animation of screen 1 start in sync with the music.

The REALST version is meant to be run at the default 72Hz monochrome mode on a real ST. This makes the animations run faster. Note that I did not test this demo on a real ST yet.

System requirements

Both versions run on default ST, as long as it has at least 1MB of RAM and a monochrome Screen. It was not written to benefit from a blitter. I have not tested the code on anything other than a default plain ST emulated in STEEM SSE.

If you want to experiment with the code, please note that from the interpreter it runs really slow (as we could see in video part 36b with the LONG{} commands), and you also need 2MB Ram at a minimum to be able to reserve enough memory.

Music

For the music I used the Yescrew/GWEM sndh player I also discussed in video part 37. With many thanks to the creators of that player. I decided to use the default song that came with it, because I liked that tune quite a lot, and for screen 3 there was hardly any CPU time left.

Code usage

Feel completely free to copy and alter the code to your liking. I created the video series and this demo to motivate everyone to code for fun. Of course, some mention when using would be highly appreciated 😉

Screens

The screens rely heavily on pre-calculations and pre-rendering to be able to animate with the desired framerates. The demo consists of 3 screens:

1) Atari ST Nostalgia sprite balls
This is just the sprite routine I used in the tutorial videos, but with the palette flipped and
A wider object to XOR over the text at the bottom. This screen runs at 36FPS on a real ST, and 30 FPS when vsyncing an emulator to 60Hz.

2) Vertical scroll with horizontal scroll text
Due to the pre-shifted screens for vertical scrolling this was a tricky one. I have to do the clean-up for the scroll-text bar twice, so there was no room left for extra animations. The scroll text uses BMOVE to move the text and RC_COPY to add the new parts. The routine is quite basic: I pre-render the entire scroll-text (so the screen preparations take some time) instead of using a mould to copy the parts from. This screen runs at 72FPS on a real ST, and 60 FPS when vsyncing an emulator to 60Hz.

3) Parallax effect with sprites and vertical scroll
This screen uses 16 pre-shifted screens for the parallax effect, just as TCB probably did in their scroller with a similar effect in the Union demo. The vertical scroll is just a BMOVE from some pre-rendered screens, that also acts as clean-up for the two sprites. This screen also runs at 72FPS on a real ST, and 60 FPS when vsyncing an emulator to 60Hz.

Other files

The following files are the version of the SNDH player I used:

Gfa_sndh.s

Gwemsndh.inl

Replay.lst

Sndhtune.inl

Please make sure to get the latest versions at https://paradize.final-memory.org/gfabasic.shtml