The major problem with the scrolling is the fact that every 8 pixels, the entire BAT table has to be updated which normally isn't a big problem but there seems to be some problem with the speed of the BAT. Only choosing to update a small section of the screen, around 10x25 characters compared to the required 45x32, still caused to redraw routine to run over the time allowed in VBLANK. The code I used was written from scratch and is the fastest it could possibly go. All the routine does is update the BAT, it doesn't use up any time calculating the tiles to draw next which will have to be there for any scrolling game. I have even tried running the code from zero page since RAM is generally faster and it produced much the same results which leads me to believe that the BAT has a serious speed problem.

Scrolling horizontally will have the exact same problems but in this case, the horizontal offset affects the tile data offset and not the screen offset so the display is corrupt without it scrolling.

Right now, what the hardware does allow is a still background using the whole display with palettes of our choosing. The drawing of this background takes a very long time however and as a result, any attempt to run games in this state will cause them to run very slowly. The lack of a 16 colour "extended" mode limits what we can do for the moment.

Since the PSG is mixing its sounds far too high, attempting to convert the VCS sounds has proved impossible as we can't tell if the envelope and noise generators are working and if they are can we just not hear them over the high pitch tone.

The lack of sprites means that virtually nothing can be tested properly.