Using The Vertical Retrace

Intro

When refreshing the video screen, if you are halfway through updating, and the refresh beam passes you up (it refreshes your new area and then starts updating the area which you haven't reached yet) you will get a screen with only half of it updated, while the other portion remains with the old data. This causes a tremendous amount of screen flicker that can really ruin things.

The answer to this problem is update our screen during a Vertical Retrace. This is when the electron beam is moving from the lower right to the upper left of the screen. Since Vram is not being accessed during this time, we can work with it at a much faster rate. Here's the code to pull it off:

The Code

```
while(inp(0x3DA) & 0x08);
while(!(inp(0x3Da)&0x08));
```

Start updating the screen!!

We are reading the Input Status Register, which is one of the video card's General Registers. We test bit 3 to see if it is set. If it is, then we know that a vertical retrace is taking place. Its just that simple. Although it looks like this will slow our routine down, it won't! This is because the video card will not insert wait states into the memory accesses that change the video memory during the retrace. We use two while's because if we detect that the video card is doing a retrace, we don't know if it is in the middle, beginning or end, so lets wait for the next one, then we'll be sure to hit it on time.

Contact Information

I just wanted to mention that everything here is copyrighted, feel free to distribute this document to anyone you want, just don't modify it! You can get a hold of me through my website or direct email. Please feel free to email me about anything. I can't guarantee that I'll be of ANY help, but I'll sure give it a try :-)

Email : deltener@mindtremors.com
Webpage : http://www.inversereality.org