Dreamcast-Talk.com

https://www.dreamcast-talk.com/forum/

Universal Deflicker/Blur Disable Code

https://www.dreamcast-talk.com/forum/viewtopic.php?t=14698

Page **1** of **2**

Universal Deflicker/Blur Disable Code

Posted: **Tue Aug 10, 2021 10:09 pm**
by **TapamN**
tl;dr: The code is

Code: Select all

```
03000001
A05F8118
0000FF00
```

It works for all games.

Most Dreamcast games ran with a deflicker filter to display a more stable image on an interlaced CRT. This works by slightly blurring the screen vertically. If you're trying to use an LCD display interlaced video, you don't really need the deflicker. This code will disable it on any game, and result in a sharper image.

It sounds like SF3 Third Strike might have had a secret option to disable it, but I'm not sure. Smash Bros Melee and Sonic Heroes had options to disable the GameCube's deflicker option.

When using VGA, most games automatically disable the deflicker, so this code won't have any effect. However, I know of three games that have deflicker on even with VGA output. Omikron, Wacky Racers, and Ready 2 Rumble (just the 1st one, not the sequel) all run with 2x horizontal supersampling, and all of these games have a vertical blur on VGA. I guess it's a mistake in the 3D driver? Using this code will remove the vertical blur while keeping 2x SSAA turned on. Fixing this was the real reason I made the code.

I actually had to disassemble Codebreaker to make this code.

The Dreamcast's 3D hardware is able to scale it's framebuffer vertically. When downscaling, there's a register that controls how to blend lines together. The deflicker works by downscaling by a single pixel (i.e. to 479), so that the filter is enabled, then having the 3D hardware vertically blur the screen when it renders. Each line is mixed with 50% of itself, 25% the line above it, and 25% the line below it. There is a hardware register that controls the percentages of the mixing.

The code sets the mixing for a line to be 99.6% itself (100% isn't possible without disabling scaling), and 0% the above and below lines. The scaling register has other functions, and changing it might have side effects on certain games. Changing the filter percentages is more likely to work with all games.

Most Codebreaker code types only allow you to modify main RAM, so it's not possible to get to the filter register. I disassembled the "in-game" part of Codebreaker to find a way to modify the register, and found two undocumented code types, not listed on the Codebreaker guide on GameFAQs. The two types of codes I found are the 03 and 08 types.

The 03 code allows writing multiple 32 bit values to anywhere, even outside of main RAM. The format looks like this:

Code: Select all

```
0300iiii
aaaaaaaa
ddddddfd (possibly multiple lines)
```

The iiii part is how many 32-bit values to write. The aaaaaaaa line is the address of where to write the values to. The dddddddd lines are what gets written there. There should be as many lines as the value of iiii. So the deflicker disable code writes one 32-bit value (0x0000FF00) to the address 0xA05F8118

There is a similar code, type 04, but that's limited to main RAM, and preforms 8-bit writes. For the 3D hardware to work correctly, it's important that a single 32-bit write is used.

The 08 code type is used to write something indirectly using a pointer. It looks like this:

Code: Select all

```
08aaaaaa
vvvvvvvv
oooooooo
```

This code reads a value from the address specified by aaaaaa, adds the oooooooo value to it, then writes vvvvvvvv to the result. The bottom two bits of the aaaaaa address control the size of the value written. If it's 0, a 8-bits are written. If it's 1, 16-bits are written. And if it's 2, 32-bits are written. This code checks that the destination is in main RAM before writing to it. If it's not, it skips the code without doing anything. I guess it's an attempt to prevent crashes from writing to a bad pointer?

In C-ish code, it looks something like this (it won't compile, but should get the point across):

Code: Select all

```
int type = aaaaaa & 0x3;
void * tmp = 0x8c000000 | (aaaaaa & 0xFFFFFC);
tmp = *(uintptr*)tmp;
tmp += oooooooo
int tmphigh =  tmp >> 24;
if (tmphigh == 0x8c || tmphigh == 0xac || tmphigh == 0x0c) {
        if (type == 0)
                *(int8_t*)tmp = vvvvvvvv;
        else if (type == 1)
                *(int16_t*)tmp = vvvvvvvv;
        else if (type == 2)
                *(int32_t*)tmp = vvvvvvvv;
}
```

That's assuming I read the disassembly correctly. It should be something along those lines.

Codebreaker seems to have an inaccessible code type. All it seems to do is skip over a number of codes without processing them.

Try changing the last line of the code to 0000A080 for a trippy effect, or 00005555 for maximum blur. It won't work if the game isn't trying to use deflicker.

Re: Universal Deflicker/Blur Disable Code

Posted: **Tue Aug 10, 2021 11:15 pm**
by **colgate**
This is impressive congrats for such dedication.

Re: Universal Deflicker/Blur Disable Code

Posted: **Tue Aug 10, 2021 11:42 pm**
by **SEGA RPG FAN**
Very cool! I didn't know it could be turned off.

Re: Universal Deflicker/Blur Disable Code

Posted: **Wed Aug 11, 2021 1:05 am**
by **Roareye**
Is there a way to patch this into the games for those of us using GDEmu?

Be useful for testing out which games it works on too, so we can list and mod the titles that need it.

Omikron: The Nomad Soul
Ready 2 Rumble
Wacky Races

Re: Universal Deflicker/Blur Disable Code

Posted: **Wed Aug 11, 2021 3:08 am**
by **TapamN**
That code doesn't patch the binary, it just has Codebreaker constantly try to set the vertical blending to do nothing.

I looked into finding a patch for GDI Wacky Racer.

I couldn't find anything that directly accessed 0xa05f8118. I guessed that there might be a function that sets PVR registers, so I looked for something that used the address 0xa05f8000. I found a function at 0x8c0f7540 that does this:

Code: Select all

```
8c0f7540:       07 d3           mov.l   0x8c0f7560,r3   ! a05f8000
8c0f7542:       3c 34           add     r3,r4
8c0f7544:       52 24           mov.l   r5,@r4
8c0f7546:       0b 00           rts
8c0f7548:       01 e0           mov     #1,r0
```

Then I looked for something that called that function with r4 set to 0x118. I found

Code: Select all

```
8c0ebbe2:       1c de           mov.l   0x8c0ebc54,r14  ! 8c0f7540
...
8c0ebc02:       1e 94           mov.w   0x8c0ebc42,r4   ! 118
8c0ebc04:       18 d3           mov.l   0x8c0ebc68,r3   ! 8c217724
8c0ebc06:       0b 4e           jsr     @r14
8c0ebc08:       32 65           mov.l   @r3,r5
```

This reads the blur setting from 0x8c217724. So I looked for where that got set.

Code: Select all

```
8c0ebd48:       db 27           or      r13,r7
8c0ebd4a:       eb 27           or      r14,r7
8c0ebd4c:       72 21           mov.l   r7,@r1
8c0ebd4e:       46 52           mov.l   @(24,r4),r2
8c0ebd50:       45 53           mov.l   @(20,r4),r3
8c0ebd52:       2c 62           extu.b  r2,r2
8c0ebd54:       2c d1           mov.l   0x8c0ebe08,r1   ! 8c217724
8c0ebd56:       18 42           shll8   r2
8c0ebd58:       3c 63           extu.b  r3,r3   !****replace with mov   #0xff, r3  -> hex ff e3****
8c0ebd5a:       2b 23           or      r2,r3           !****replace with shll8        r3      -> hex 18 43****
8c0ebd5c:       32 21           mov.l   r3,@r1
8c0ebd5e:       43 52           mov.l   @(12,r4),r2
8c0ebd60:       01 e5           mov     #1,r5
8c0ebd62:       47 50           mov.l   @(28,r4),r0
8c0ebd64:       44 53           mov.l   @(16,r4),r3
8c0ebd66:       59 22           and     r5,r2
8c0ebd68:       59 20           and     r5,r0
```

That appears to be the only place that writes to 0x8c217724, unless there's something that does it indirectly, but I doubt it.

Patching the marked lines of WACKY.BIN should always disable the vertical blur. To convert the address listed in the left column of the disassembly to a position in WACKY.BIN, subtract 0x8c010000 from it.

This causes some bits in the blend register that are (probably) unused to be set to ones, but it should be ok.

Not sure what the best way to patch a GDI would be. I guess a brute force way would be to look through track03.bin for something that looks similar and replace 3c 63 2b 23 with ff e3 18 43 and hope it works? It looks like there are multiple locations in track03.bin where the surrounding code shows up.

I don't have a way to test this myself.

Re: Universal Deflicker/Blur Disable Code

Posted: **Wed Aug 11, 2021 7:06 am**
by **Espiral**
I can confirm that changing 3c 63 2b 23 with ff e3 18 43 does the work, the game looks noticeable crisper specially in person and in motion.

original
Image

deblur

Image

BTW is the game rendering internally at 1280*480 or 640*960? I say this because the super sampling (AA) is noticeable on verticar lines, not horizonatl ones, shouln't that mean that the game is rendering double the resolution vertically instead of horizontally?

Re: Universal Deflicker/Blur Disable Code

Posted: **Wed Aug 11, 2021 4:53 pm**
by **TapamN**
It's rendering at 1280x480. Increasing the horizontal resolution helps with vertical edges.

I looked at Omikron and R2RB, and the section of code for 3D driver that needs to be patched seems to be identical, just a different location (8c0d5b98 for Omikron, and 8c0dd418 for R2RB), so the same trick should work for those.

I forgot to mention this, but the reason I posted so much surrounding code in my previous post was to help people identify the correct place to patch. Instead of searching for just 3c 63 2b 23, you could check at the surrounding bytes to make sure it's the same code and not something that just happens to have those 4 bytes.

Re: Universal Deflicker/Blur Disable Code

Posted: **Thu Aug 12, 2021 6:39 am**
by **Espiral**
Any other games that render at 1280x480?
I think I know the answer for this one but, could we force more games to render at 1280*480?

Re: Universal Deflicker/Blur Disable Code

Posted: **Thu Aug 12, 2021 7:58 am**
by **mstar**

One of the most impressive things i've read in a while!

That's true dedication (and maybe being a little bit nuts)

Keep it up!

Re: Universal Deflicker/Blur Disable Code

Posted: **Thu Aug 12, 2021 12:29 pm**
by **ChristinaDreamcustDC**

> *TapamN wrote:*tl;dr: The code is
>
> Code: Select all
>
> ```
> 03000001
> A05F8118
> 0000FF00
> ```
>
> It works for all games.
>
> Most Dreamcast games ran with a deflicker filter to display a more stable image on an interlaced CRT. This works by slightly blurring the screen vertically. If you're trying to use an LCD display interlaced video, you don't really need the deflicker. This code will disable it on any game, and result in a sharper image.
>
> [...]

This is similar to development in N64 circles. Thanks for your hard effort!
