

## Cemu - Bug #304

### Problems in Bicubic&Hermite scaling filters

04/03/2020 01:24 AM - NAVras

<b>Status:</b>	New	<b>Start date:</b>	04/03/2020
<b>Priority:</b>	Normal		
<b>Assignee:</b>			
<b>Category:</b>	Graphic		
<b>API:</b>	OpenGL	<b>GPU Vendor/Model:</b>	
<b>Cemu Version:</b>	1.17.4		
<b>Description</b>			
<p>They were found half a year ago but I forgot to submit a ticket (for some reason...?). Sorry if I misremember or provide outdated information.</p>			
<p>1. <math>f_x</math> &amp; <math>f_y</math> in Bicubic "<math>f = \text{fract}(\text{texcoord})</math>" only makes sense if the sampling point at "passUV" uses top-left instead of center as its coordinate. The long version: cubic() outputs the weight of 4 texels (one axis) in B-Spline with the assumption of them being <math>d-1, d_0, d+1, d+2</math> (see <a href="https://vec3.ca/bicubic-filtering-in-fewer-taps/">https://vec3.ca/bicubic-filtering-in-fewer-taps/</a> "The B-Spline Weighting Function") <math>d_0</math> is always the texel on the left side of the sampling point so that the 4 texels are the ones whose middle point is the closet to the sampling point. Input <math>f</math> is the distance of the sampling point from <math>d_0</math>. The correct <math>f</math> is shown both in above website and <a href="https://github.com/slashiee/cemu_graphic_packs/blob/master/Filters/Bicubic%20(4%20taps)/output.glsl#L23-L29">https://github.com/slashiee/cemu_graphic_packs/blob/master/Filters/Bicubic%20(4%20taps)/output.glsl#L23-L29</a> (If texcoord were top-left, "<math>\text{fract}(\text{texcoord})</math>" would indeed be the distance by pretending texel to be top-left as well.)</p>			
<p>2. "uniform vec2 textureSrcResolution" gives the native resolution of the game, doesn't reflect the actual texture resolution if it has been changed by gfx pack rule. Hermite is now using "uniform vec2 outputResolution", which gives the actual output resolution not the "texelSize" it requires. "texelSize" is used to turn "passUV" from 0--1 to 0--texelSize, which is then used to locate texels or apply offsets conveniently, native/output resolution are not involved in this.</p>			
<p>3. Final textures to be resized have <code>gl_texture_wrap_* = GL_REPEAT</code> (BoTW's main menu, only menu not in-game), <math>d+1</math> <math>d+2</math> for pixels at the edge for example can be texels at other side of the texture, which causes a bright border ( <a href="https://user-images.githubusercontent.com/21133512/62403404-7b170d00-b5bf-11e9-957e-f7c4baa07f02.PNG">https://user-images.githubusercontent.com/21133512/62403404-7b170d00-b5bf-11e9-957e-f7c4baa07f02.PNG</a>). (Also Bug #253) If it can be overwritten <code>GL_MIRRORED_REPEAT</code> would be a good choice.</p>			