MXWendler: Developing Shaders Version 1.2

This document describes the MXWendler (MXW) GLSL effect interface. It references writing and modifying MXWendler FXPack files, valid for builds > 5000

Contents

What is a FXPack shader?

A GLSL shader is a scripted effect program, which is compiled in runtime into a GPU binary. It will be massively parallel executed by the GPU during runtime. With these shaders, you can realize a lot of visually pleasing video effects.

Why should i write or modify MXWendler FXPack files?

There is a collection of already existing 150 effects. For some visual ideas, you have to create your own or modify existing ones.

How do i create or modify FXPack files?

FXPack files exist in the Program / effects folder. They are named .fxpack, but in fact they are zip files. Either rename to shader.zip or shader.fxpack.zip and open with e.g. 7-Zip FM (Filemanager) or do command-line edit with vim.

What are the UI Interfaces into shader uniforms?

`uniform float mxw_millis_mxw`

> This uniform will receive the current application runtime in milliseconds

`uniform float mxw_framecounter_environment_mxw`

> This uniform will receive the current application runtime in frames (no fraction)

`uniform float mxw_maxU_mxw`

> This uniform will receive maximal U value of the current footage in 0..1 (only used in MXW versions < 3.0, now always 1.0)

`uniform float mxw_maxV_mxw`

> This uniform will receive maximal V value of the current footage in 0..1 (only used in MXW versions < 3.0, now always 1.0)

`uniform float mxw_reciprocalU_mxw`

> This uniform will receive maximal reciprocal U value of the current footage in 0..1 (only used in MXW versions < 3.0, now always 1.0)

`uniform float mxw_reciprocalV_mxw`

> This uniform will receive maximal reciprocal V value of the current footage in 0..1 (only used in MXW versions < 3.0, now always 1.0)

`uniform float mxw_framenumber_mxw`

> This uniform will receive the current clip position as a float – e.g. 3.5 means in the middle between frame 3 and 4, useful for e.g. frame blending actions

`uniform float mxw_framecounterenvironment_mxw`

> This uniform will receive the current frame counter count as it is available in mxw.framecounter in javascript.

`uniform float mxw_footagesize_mxw`

> This uniform will receive the footage/rendering size in X

```
uniform float mxw_footagesizeY_mxw
```

> This uniform will receive the footage/rendering size in Y

```
uniform float mxw_footageframelength_mxw
```

> This uniform will receive the footage duration in frames (1 for FinalFX)

```
uniform float mxw_viewportsizeX_mxw
```

> This uniform will receive the current output window size in X

```
uniform float mxw_viewportsizeY_mxw
```

> This uniform will receive the current output window size in Y

```
uniform float mxw_outputLuminance1x1_mxw
```

> This uniform will receive the current output rendered into a 1x1 pixel, thus showing the current general brightness

```
uniform sampler2D mxw_tex_minus_0_mxw
```

> This uniform will receive a handle to the current video texture

```
uniform sampler2D mxw_tex_minus_1_mxw
```

> This uniform will receive a handle to the previous (current-1) video texture

```
uniform float mxw_vertslider_Master_0x0_1x0_1x0_mxw
```

> This uniform will create a slider with the name ‚Master', with the lowest value of 0.0, the highest value of 1.0 and an normalized initial value of 1.0

```
uniform float mxw_vertslider_scale_0x0_100x0_1x0_mxw
```

> This uniform will create a slider with the name ‚scale', with the lowest value of 0.0, the highest value of 100.0 and an normalized initial value of 1.0 (=100)

```
uniform sampler2D mxw_tex_ag_ascii_jpg_mxw
```

> This uniform will receive a handle to a picture file named ‚ag_ascii.jpg' found in the ‚textures' folder in the .fxpack file

```
uniform vec4 mxw_colorcontrol_Fxgain_0x5_0x5_0x5_1x0_mxw
```

> This uniform will receive rgba values from a color control named ‚Fxgain'

11.24 add mxw_framenumberenvironment_mxw