PowerVR
by Imagination

PFX Language Format

Specification

Filename	:	PFX Language Format.Specification
Version	:	PowerVR SDK REL_3.2@2607465a External Issue
Issue Date	:	10 Sep 2013
Author	:	Imagination Technologies Limited

Contents

List of Figures

No table of figures entries found.

1. Introduction

1.1.   Document Overview

The purpose of this document is to act as a specification for the PowerVR Effects (PFX) format.

1.2.   Specification Version

PFX Specification Version:

   2.0.2

Changes from previous revision:

- Added the 'Changes from previous revision' section.
- Updated Section 3.2.1 & Section 3.3.1. Tables previously stated 'MIP-MAP' and should have stated 'MIPMAP'.

2. PFX Overview

The PFX format is a small, simple, easy to use effects format consisting of several blocks that describe how a given graphics effect is put together (see Section 3 Blocks). As a minimum, a correctly formatted PFX consists of:

- One 'EFFECT' block
- One 'VERTEXSHADER' block
- One 'FRAGMENTSHADER' block

It is also possible for PFXs to contain the following:

- One 'TARGET' block
- Zero or more 'TEXTURE' blocks

By default PFXs are stored in .pfx files. It is possible for multiple PFXs to exist within a single .pfx file, each described by a separate effect block; in this instance multiple PFXs may share blocks.

Finally, it is possible for a PFX to reference a 'TARGET' block as an input as if it were a 'TEXTURE' block, enabling the simple creation of complex post-processing effects. For this to function correctly the 'TARGET' block render should be completed prior to being read as an input. If the 'TARGET' block render has not been completed prior to being read as an input, the behaviour will vary based on the render target implementation of the platform.

3. Blocks

3.1. HEADER

The header block contains metadata used for description and labelling purposes.

3.1.1. Keywords

.

Keyword	Description
VERSION	The version of this PFX. Format is: MAJOR.MINOR.BUILD.REVISION
DESCRIPTION	A plain-text description of what this PFX file contains and the desired effect.
COPYRIGHT	Copyright descriptor of author(s).

3.2. TEXTURE

A 'TEXTURE' block describes a surface that can either be the contents of a texture file, or the contents of a frame buffer/render texture.

3.2.1. Keywords

Keyword	Description
NAME	A text identifier for this texture.
PATH	The filename of the texture. If spaces are included, enclose the path in quotation marks. E.g. `"base map.pvr"`.
MINIFICATION	Minification texture filter flags. Valid values: • NEAREST • LINEAR
MAGNIFICATION	Magnification texture filter flags. Valid values: • NEAREST • LINEAR
MIPMAP	MIP-map texture filter flags. Valid values: • NEAREST • LINEAR • NONE
VIEW	Specifying the VIEW keyword modifies the functionality of the defined texture to be a render texture of the current scene. Valid values: • PFX_CURRENTVIEW • POD camera name in optional quotation marks. E.g. `"Camera01"`.
CAMERA	An alias for VIEW.
RESOLUTION	Describes the resolution of the texture. This will be ignored if PATH is specified – the resolution will be that of the loaded texture.
WRAP_*x*	Where *x* is a valid axis (S, T, or R). Specifies the texture wrapping in the defined axis. Valid values: • REPEAT • CLAMP

Keyword	Description
SURFACETYPE	Describes the surface/pixel type of the texture. This will be ignored if PATH is specified – the surface type will be that of the loaded texture. Valid values: • RGBA8888 • RGBA4444 • RGB888 • RGB565
FILTER	Deprecated. Allows the specification of texture filter flags in short-hand. Valid values are hyphen separated filter flags in the order of minification, magnification, and MIP-map. E.g. LINEAR-LINEAR-LINEAR enables trilinear texture filtering, or LINEAR-LINEAR-NONE enables bilinear.
WRAP	Deprecated. Allows the specification of texture wrapping flags in short-hand. Valid values are hyphen separated wrapping flags in the order of axis S, T, and R. E.g. REPEAT-CLAMP-CLAMP.

3.2.2.   Values

Value	Associated Keyword	Description
PFX_CURRENTVIEW	VIEW CAMERA	Indicates that the render texture should be derived from the current view of the scene, and not from a specific POD file camera.
LINEAR	MINIFICATION MAGNIFICATION MIP-MAP	Linearly interpolates between sampled texels.
NEAREST		Chooses nearest texel based on Manhattan distance.
NONE	MIP-MAP	Disable MIP-mapping.
CLAMP	WRAP_S WRAP_T WRAP_R	Clamps to texture border.
REPEAT		Repeats at texture border.
RGBA8888	SURFACETYPE	32bit RGBA texture format.
RGBA4444		16bit RGBA texture format.
RGB888		24bit RGB texture format.
RGB565		16bit RGB texture format.
INTENSITY8		8bit intensity texture format.

3.3.   TARGET

A 'TARGET' block specifies a surface that an 'EFFECT' block can render to. A PFX may read from a 'TARGET' block as if it were a 'TEXTURE' block as long as the render to that block has been completed prior to the read.

3.3.1.   Keywords

Keyword	Description
NAME	A text identifier for this target.
MINIFICATION	Minification texture filter flags. Valid values: - NEAREST - LINEAR
MAGNIFICATION	Magnification texture filter flags. Valid values: - NEAREST - LINEAR
MIPMAP	MIP-map texture filter flags. Valid values: - NEAREST - LINEAR - NONE
RESOLUTION	Describes the resolution of the texture. This will be ignored if PATH is specified – the resultant resolution will be that of the loaded texture.
WRAP_x	Where x is a valid axis (S, T, or R). Specifies the texture wrapping in the defined axis. Valid values: - REPEAT - CLAMP
SURFACETYPE	Describes the surface/pixel type of the texture. This will be ignored if PATH is specified – the resultant surface type will be that of the loaded texture. Valid values: - RGBA8888 - RGBA4444 - RGB888 - RGB565

3.3.2. Values

Value	Associated Keyword	Description
PFX_CURRENTVIEW	VIEW CAMERA	Indicates that the render texture should be derived from the current view of the scene, and not from a specific POD file camera.
LINEAR	MINIFICATION MAGNIFICATION MIP-MAP	Linear interpolates between sampled texels.
NEAREST		Chooses nearest texel based on Manhattan distance.
NONE	MIP-MAP	Disable MIP-mapping.
CLAMP	WRAP_S WRAP_T WRAP_R	Clamps to texture border.
REPEAT		Repeats at texture border.
RGBA8888	SURFACETYPE	32bit RGBA texture format.
RGBA4444		16bit RGBA texture format.
RGB888		24bit RGB texture format.
RGB565		16bit RGB texture format.
INTENSITY8		8bit intensity texture format.

3.4.   VERTEXSHADER / FRAGMENTSHADER

'VERTEXSHADER' and 'FRAGMENTSHADER' blocks are the location in which the GLSL code for an effect is located. The code can either be embedded within the PFX file itself, or be located elsewhere and referenced with a filename using the 'FILE' keyword.

3.4.1.   Keywords

Keyword	Description
NAME	A unique identifier for this shader.
FILE	A file name of a text file containing valid GLSL code.

3.4.2.   Sub-blocks

Block	Description
GLSL_CODE	A block containing plain-text GLSL code.

3.5.   EFFECT

The 'EFFECT' block is the primary block used in describing a PFX; it references other blocks which can contain textures, targets, and shaders as well as containing a number of application specific 'semantics' which can be used by an application to identify the meaning of a given attribute.

3.5.1.   Keywords

Keyword	Description
NAME	A text identifier for this effect.
ATTRIBUTE	Specifies GLSL 'attribute' variable. Format: `ATTRIBUTE    varName    SEMANTIC` • 'varName' references a variable as specified in the shader blocks. • 'SEMANTIC' references an application specific semantic.
UNIFORM	Specifies GLSL 'uniform variable. Format: `UNIFORM    varName    SEMANTIC` • 'varName' references a variable as specified in the shader blocks. • 'SEMANTIC' references an application-specific semantic.
TEXTURE	Specifies a texture name which will be bound to the given unit. Format: `TEXTURE    UNIT    TextureName` • 'UNIT' specifies an integer-based texture unit to bind to. • 'TextureName' references a 'TEXTURE' block of a given name.
TARGET	Specifies a target which this effect will write to, instead of the default frame buffer. Two types of targets exist, colour targets, and depth targets. Only colour target support is required to be compliant with this specification. Format: `TARGET    BUFFERTYPE<UNIT>    TargetName sdfsdf` • 'BUFFERTYPE' can be of type 'COLOR' or, optionally, 'DEPTH'. • 'UNIT' is an integer based value defining the buffer unit. Only '0' is required to be compliant with this specification. • TargetName references a 'TARGET' block of a given name.
VERTEXSHADER	References a VERTEXSHADER block of a given name.
FRAGMENTSHADER	References a FRAGMENTSHADER block of a given name.

3.5.2.   Sub-blocks

Block	Description

Block	Description
`ANNOTATION`	A block containing plain-text which will be copied as-is into a text buffer, readable by the application.

3.6.   Deprecated Blocks

The following blocks are now deprecated and will be removed in a future version of the specification, their use is discouraged.

3.6.1.   TEXTURES

The 'TEXTURES' block has been replaced with multiple 'TEXTURE' blocks; this allows the specification of individual textures in a more verbose manner.

Each line of the 'TEXTURES' block takes the form of:

```
FILE     TextureName     FileName.pvr    FILTERFLAGS WRAPFLAGS
```

Keywords

Value	Description
TextureName	Specifies a text identifier.
FileName.pvr	Specifies the filename of the texture.
FILTERFLAGS	Specifies a hyphen separated list of texture filter flags. See Section 3.2.1.
WRAPFLAGS	Specifies a hyphen separated list of texture wrapping flags. See Section 3.2.1.

4. Contact Details

For further support contact:

devtech@imgtec.com

PowerVR Developer Technology
Imagination Technologies Limited
Home Park Estate
Kings Langley
Herts, WD4 8LZ
United Kingdom

Tel: +44 (0) 1923 260511
Fax: +44 (0) 1923 277463

Alternatively, you can use the PowerVR Insider forums:

www.imgtec.com/forum

For more information about PowerVR or Imagination Technologies Limited visit our web pages at:

www.imgtec.com