NVIDIA OpenGL Extension For G8x
EXT_DRAW_BUFFERS
and
NV_FRAGMENT_PROGRAM4

김혁

2008.03.28

EXT_DRAW_BUFFERS2

- OpenGL 2.0 is required
- Provides seperate blend anbles
  and color write masks for each color output.

Introduction to DrawBuffer

Supported Draw Buffers

- GL_FRONT, GL_BACK
- GL_FRONT_AND_BACK
- GL_FRONT_LEFT, GL_FRONT_RIGHT
- GL_BACK_LEFT, GL_BACK_RIGHT
- GL_AUXi
- GL_NONE

Using Draw Buffers

Sample Code in C/C++

```
glDrwaBuffer( GL_FRONT );
   or
glDrawBuffers( 2, buffers );
```

Sample Code in GLSL

```
gl_FragColor = FinalColor;
   or
gl_FradData0 = FinalColor
gl_FradData1 = FinalColor
```

New Functions for Enabling/Disabling in EXT_DRAW_BUFFERS2

- void EnabledIndexedEXT(enum target, uint index);
- void DisabledIndexedEXT(enum target, uint index);
- boolean IsEnabledIndexedEXT(enum target, uint index);

  Eg.) For GL_DRAW_BUFFER3
  bEnabled = IsEnabledIndexedEXT(GL_BLEND, 3);

Overview of the Indexed Blending

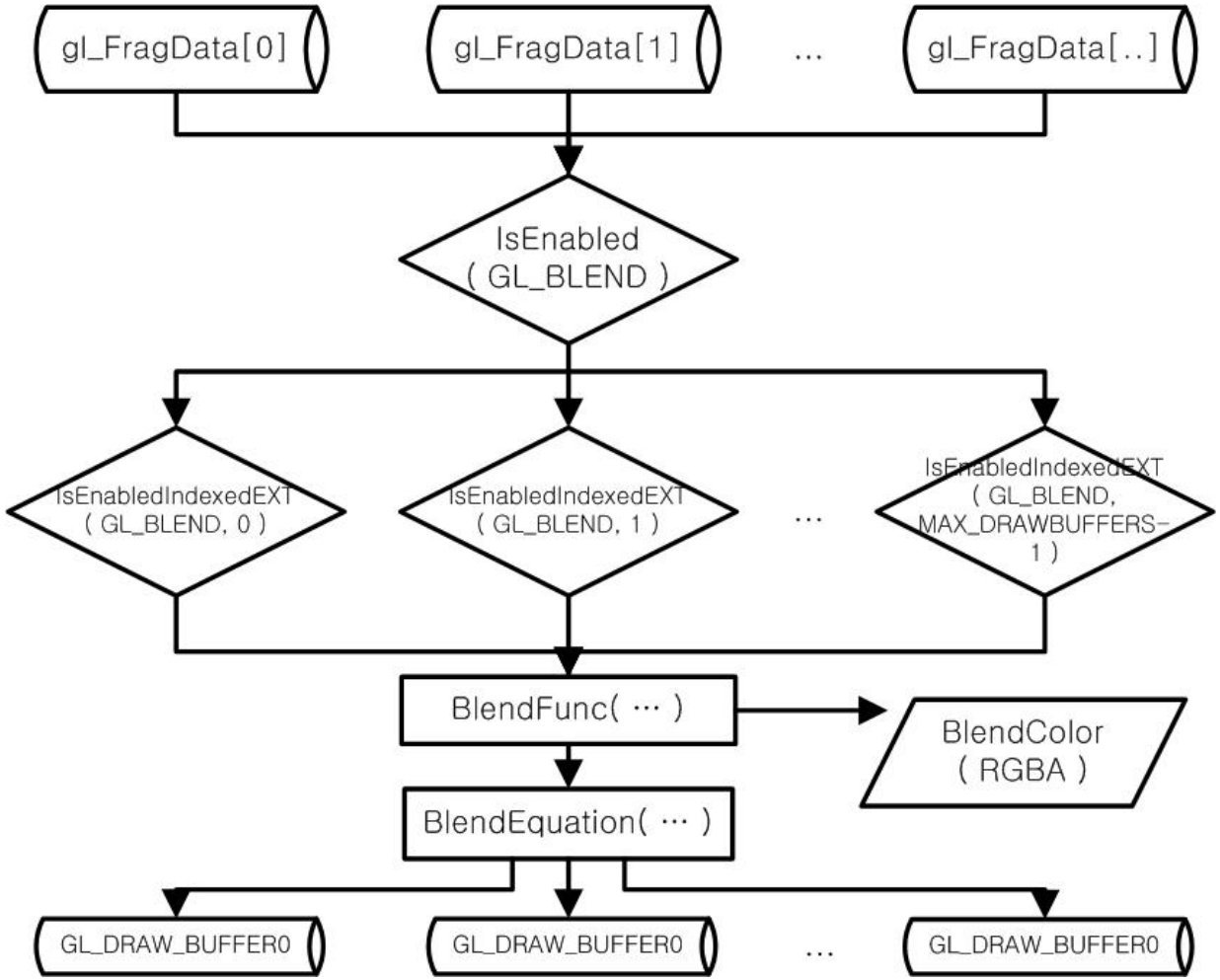

Figure: Indexed Blending

New Functions for Mask

- Mask for color index mode
  - void IndexedMaskEXT(uint mask);
    Each bit refers to corresponding color index buffer.

- Color mask for RGBA mode
  - void ColorMask(boolean r, boolean g, boolean b, boolean a);
  - void ColorMaskIndexedEXT(uint buf, boolean r, boolean g, boolean b, boolean a);

New Fuinctions of Querying Data

- void GetIntegerIndexedvEXT(enum target, uint index, int *data);
- void GetBooleanIndexedvEXT(enum target, uint index, boolean *data);

```
Eg.) For GL_DRAW_BUFFER0
  GetBooleanIndexedvEXT(
    GL_COLOR_WRITEMASK, 0, &bEnabled );
  GetBooleanIndexedv(
    GL_DRAW_BUFFER0_COLOR_WRITEMASK_EXT,
    &bEnabled );
```

NV_FRAGMENT_PROGRAM4

- OpenGL 1.1 is required
- NV_GPU_PROGRAM4 is required
- This extension builds on the common assembly instruction set.
- ARB_FRAGMENT_PROGRAM can be ported by changing the program header "!!ARBfp1.0" to "!!NVfp4.0".
  There are small number of unported instructions in this way.

Interpolation Modifier

- FLAT
  For flat-shading
- CENTROID
  Which interpolates using a point on or inside primitive
- NOPERSPECTIVE
  linear interpolation in screen space

Sample Code in Fragment Program

Eg. 1)

```
CETROID ATTRIBUTE texCoord;
```

Eg. 2)

```
CETROID NOPERSPECTIVE ATTRIBUTE linearColor;
```

CENTROID Interpolation Modifier

Problem of Non-CENTROID Sampling [1]

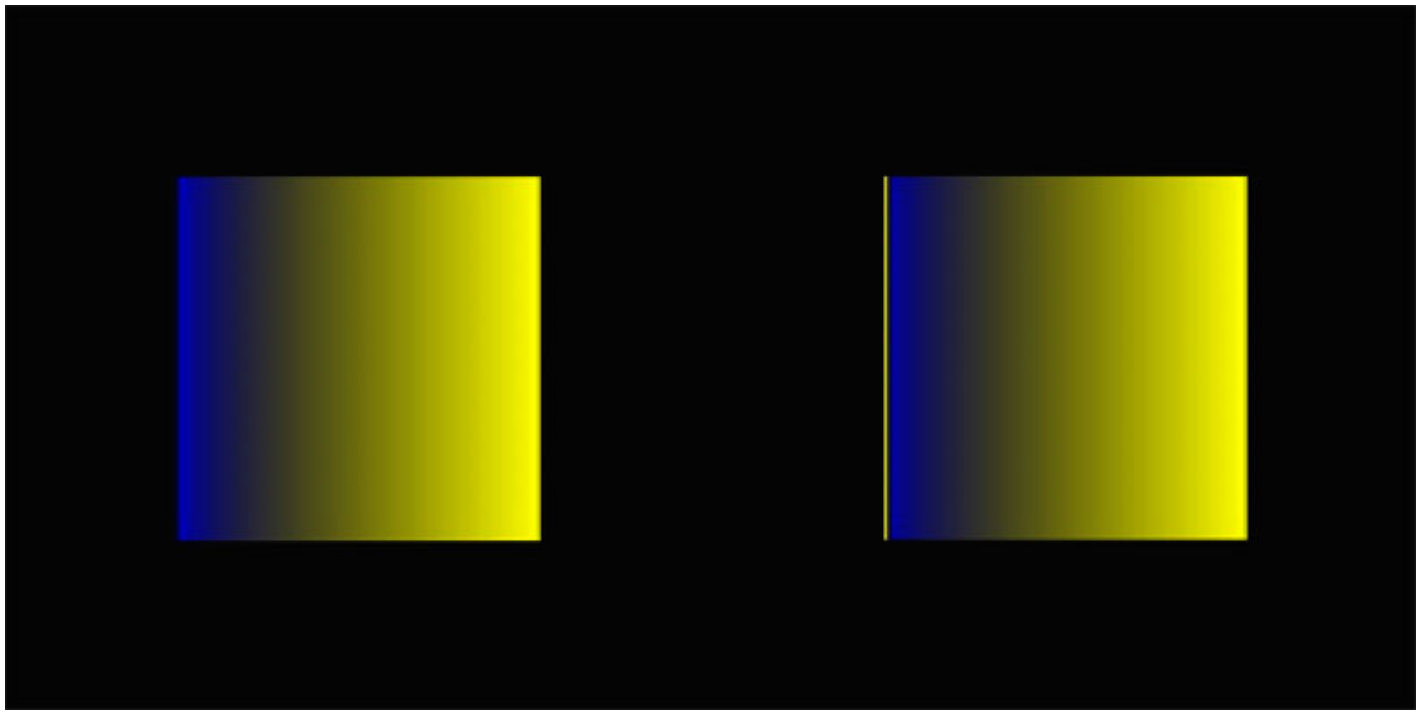

Figure: Problem in Multi-sampling

[1] GLSL: Center or Centroid?

CENTROID Interpolation Modifier

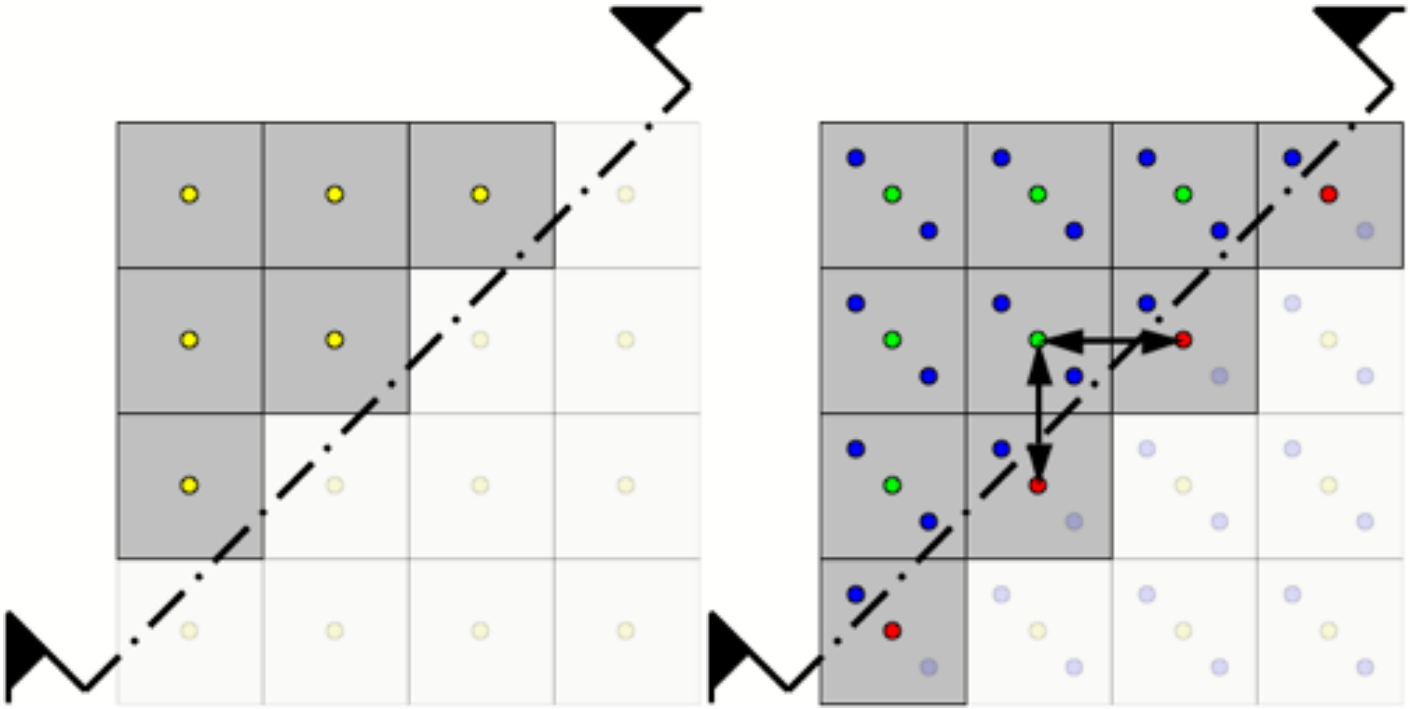

Figure: Single/Multi-sampling

- No problem with single sample rasterization
- In multi-sampling, interpolation value can be out of range.

NOPERSPECTIVE Interpolation Modifier

- Perspective Equations[2]

$$f = \frac{(1-t)\frac{f_a}{w_a} + t\frac{f_b}{w_b}}{\frac{(1-t)}{w_a} + \frac{t}{w_b}} \tag{1}$$

$$f = \frac{a\frac{f_a}{w_a} + b\frac{f_b}{w_b} + c\frac{f_c}{w_c}}{\frac{a}{w_a} + \frac{b}{w_b} + \frac{c}{w_c}} \tag{2}$$

- Non-perspective Equation(approximation to eq.2)

$$z = (1-t)z_a + (t)z_b \tag{3}$$

[2] 3.4.1 and 3.4.2, OpenGL 2.1 Specification

Notes and Restrictions in Interpolation Modifier

- No effect on point primitives, drawpixels, bitmap
- Must be FLOAT variables
  (INT or UINT must be float)
- Only for ATTRIBUTE variables
- FLAT modifier cannot be used with others
- Variables must not be bound to fragment's position, face direction, fog coordinate, or any interpolated clip distance
- Attribute variables with different modifiers must not be bound to the same fragment attribute
- Frangment's primary color and secondary color must be of same modifier

Fargment Attribute Bindings

	Fragment Attribute Binding	Components
	fragment.color	(r,g,b,a)
	fragment.color.primary	(r,g,b,a)
	fragment.color.secondary	(r,g,b,a)
	fragment.texcoord	(s,t,r,q)
	fragment.texcoord[n]	(s,t,r,q)
	fragment.fogcoord	(f,-,-,-)
$*$	fragment.clip[n]	(c,-,-,-)
	fragment.attrib[n]	(x,y,z,w)
	fragment.texcoord[n..o]	(s,t,r,q)
$*$	fragment.clip[n..o]	(c,-,-,-)
	fragment.attrib[n..o]	(x,y,z,w)
$*$	fragment.position	(x,y,z,1/w)
$*$	fragment.facing	(f,-,-,-)
$*$	primitive.id	(id,-,-,-)

Interpolation modifiers are not supported on variables with "$*$".

Result Bindings

Binding	Comonents	Description
result.color	(r,g,b,a)	color
result.color[n]	(r,g,b,a)	color output n
result.depth	(*,*,d,*)	depth coordinate

- If geometry program is enabled, primitive ID is not available automatically to the fragment program.
  To use that, use 'result.primid' in geometry program. And primitive.id of emitted primitives is same as id of provoking primitive.

Other Issues

- Fixed-function Fog Emulation

  `ARB_FOG_EXP, ARB_FOG_EXP2, ARB_FOG_LINEAR`

- KIL : Kil Fragmen

  KIL is available only to fragment programs.

- Interpolation mode in fixed-function state

```
glInterpolateAttribNV( GL_TEXTURE0, GL_FLOAT );
glInterpolateAttribNV( GL_GENERIC_ATTRIB0, GL_CENTROID_NV );
```