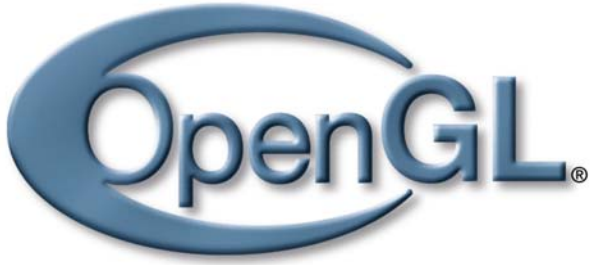
Cool GLSL Tips

Bill Licea-Kane

© Copyright Khronos Group, 2007 - Page



Cool GLSL Tips

Bill Licea-Kane

AMD

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips

- Contributions to arb-glsl workgroup
  - NOT final, subject to (minor?) change
- Not exhaustive
  - There is more cool stuff that will not be covered today

CAVEAT!

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips

- Interpolation Control
- Integer Support
- New Texture
- Plus more we won't get to today

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid

- Already in version 120

```
#version 120
centroid varying float myMixer;
```

© Copyright Khronos Group, 2007 - Page

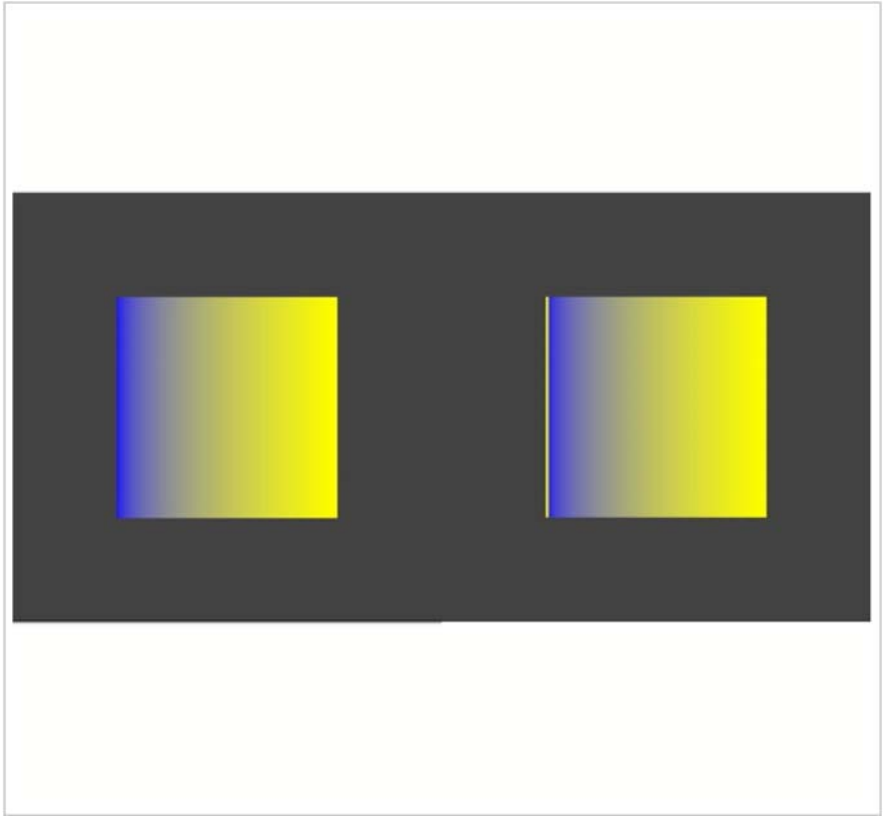
Cool GLSL Tips - Centroid

```
varying float myMixer;
// Interpolate color between blue and yellow.
// Let's do a sqrt for a funkier effect.
void main( void )
{
    const vec3 blue   = vec3( 0.0, 0.0, 1.0 );
    const vec3 yellow = vec3( 1.0, 1.0, 0.0 );

    float a = sqrt( myMixer );
    vec3 color = mix( blue, yellow, a ); // nonlerp
    gl_FragColor = vec4( color, 1.0 );
}
```

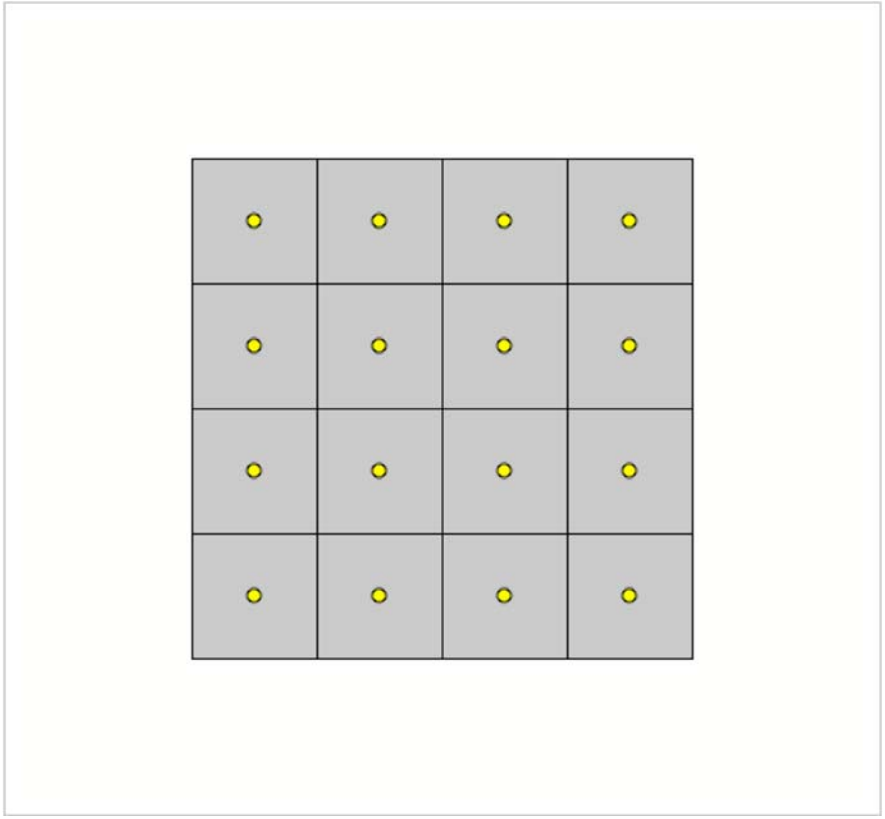
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



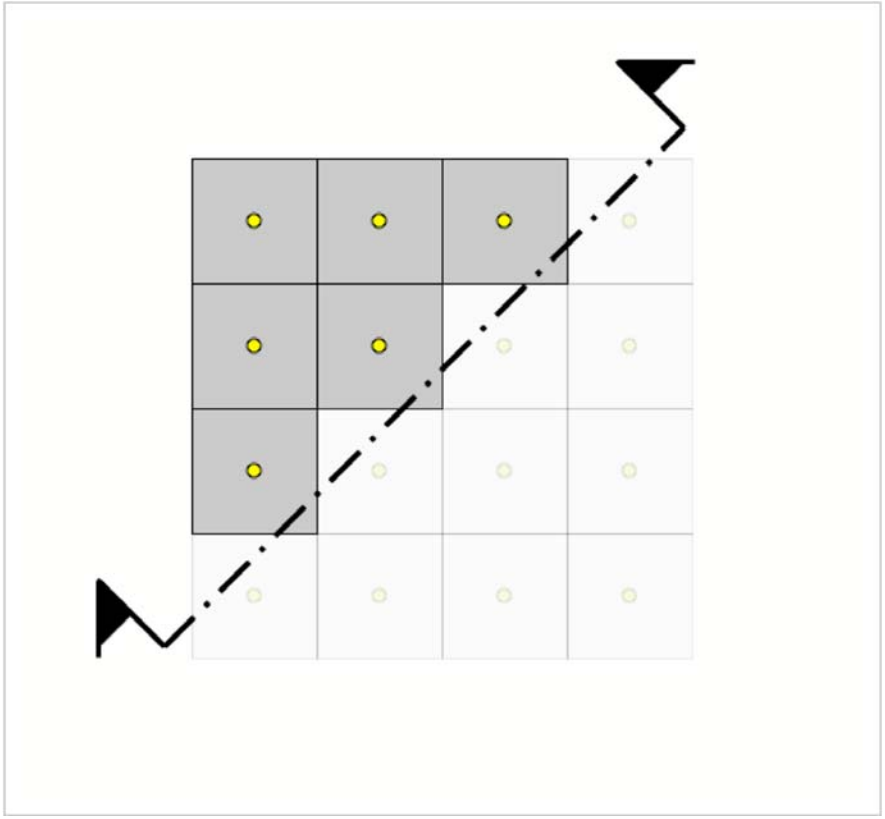
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



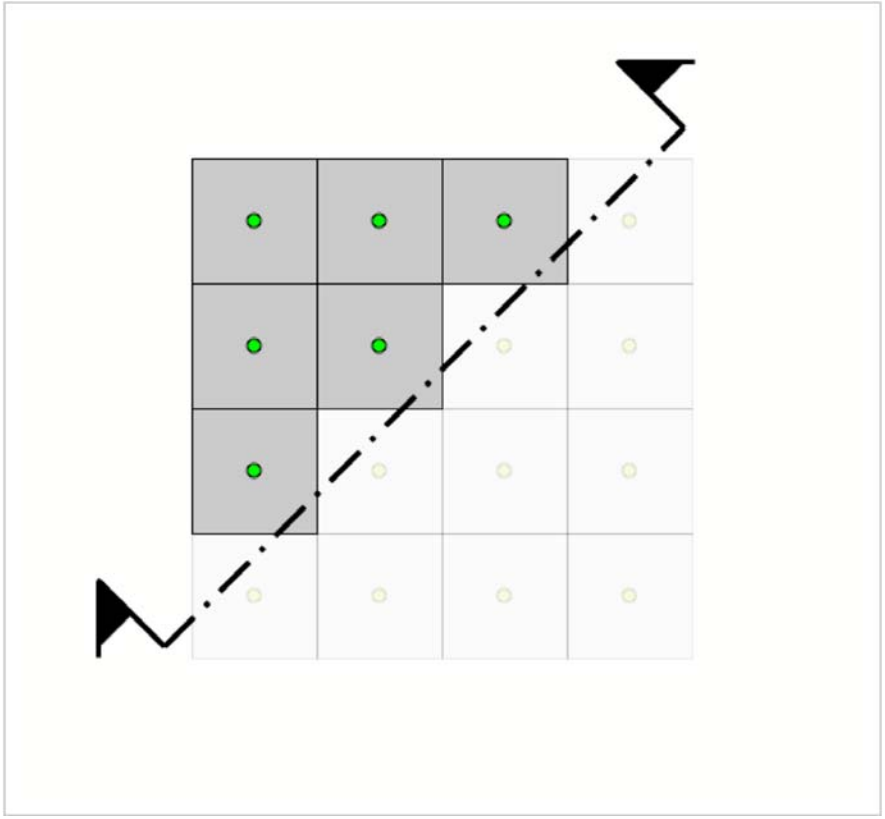
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



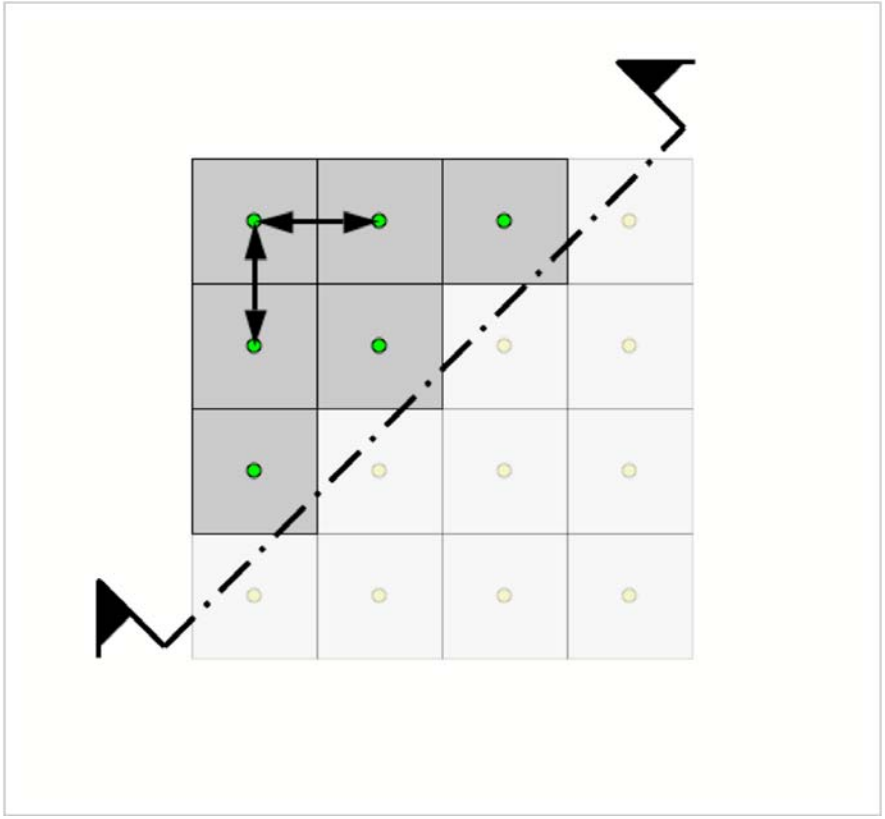
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



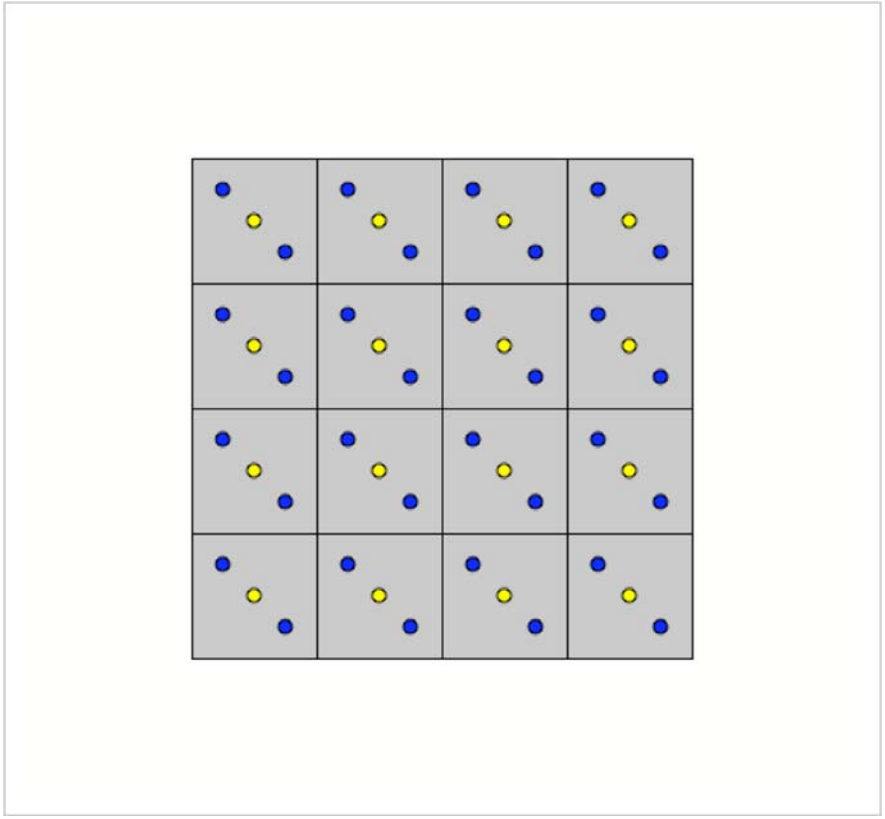
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



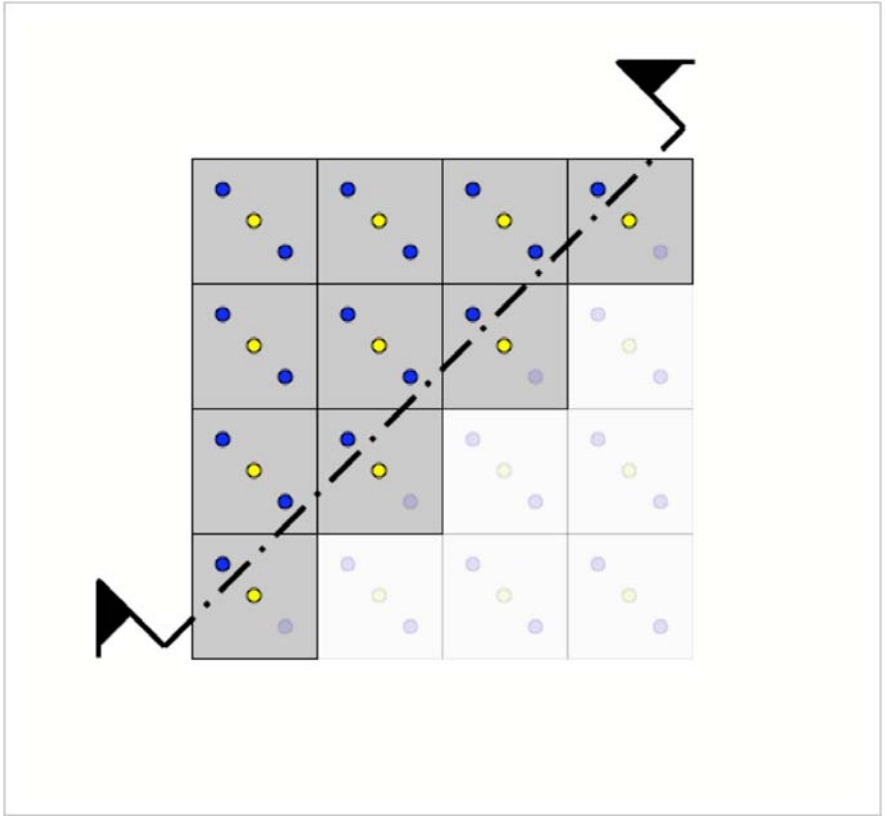
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



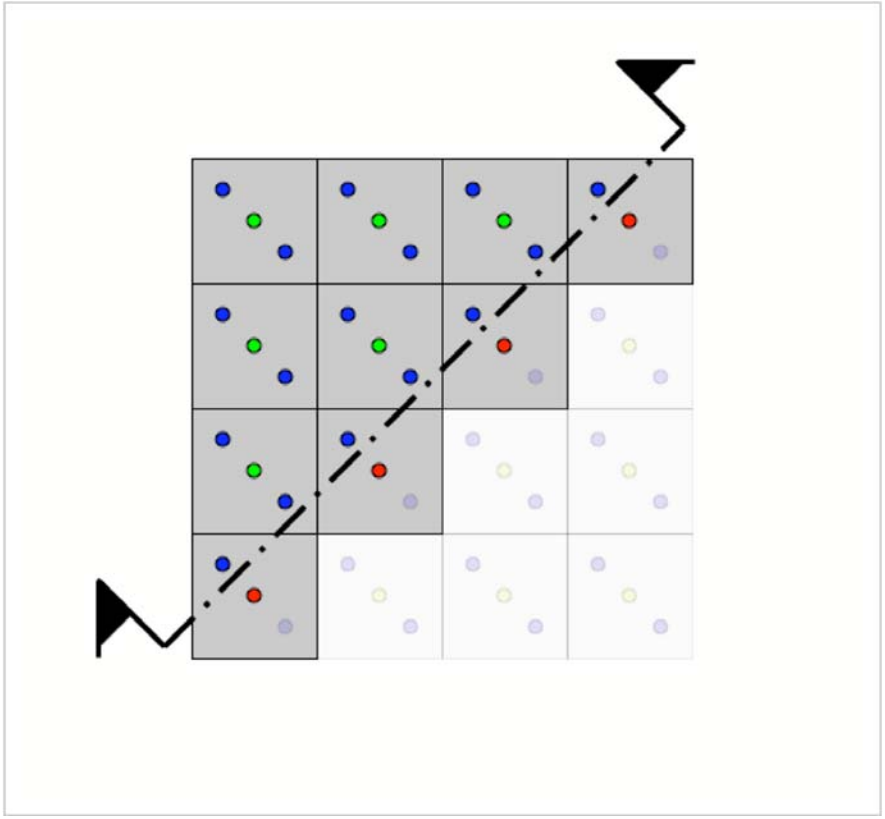
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



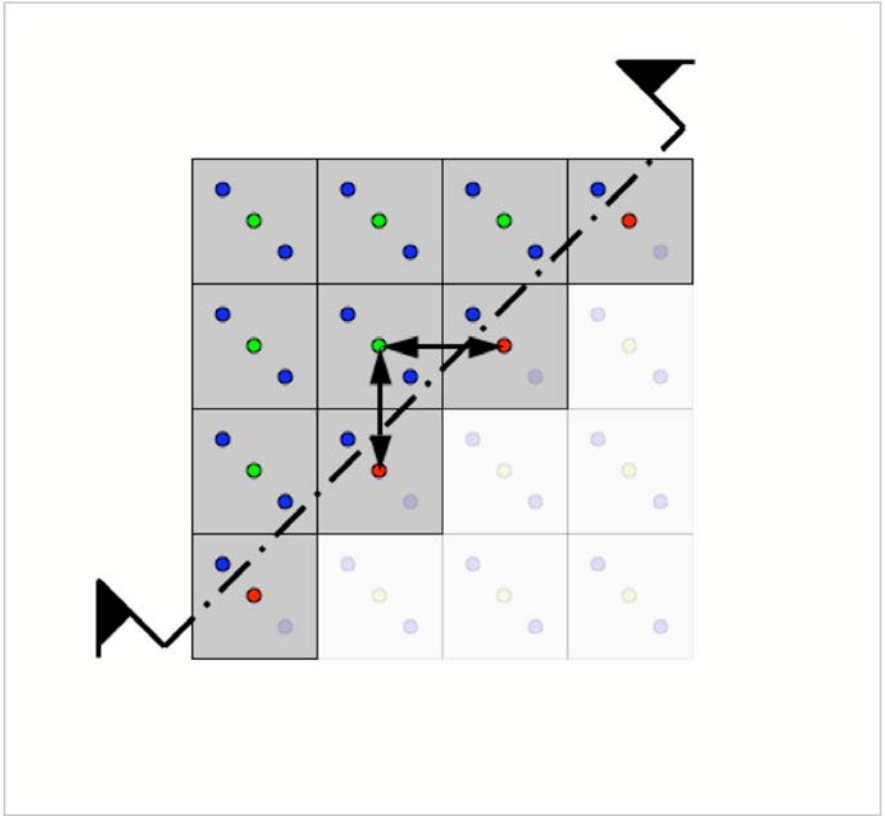
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



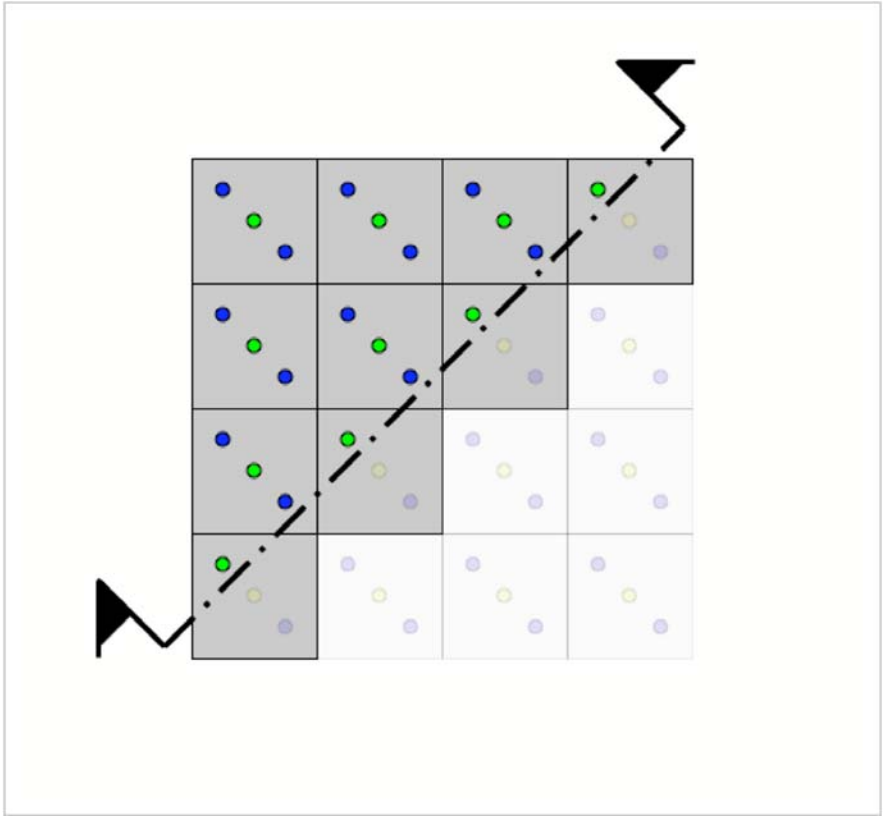
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



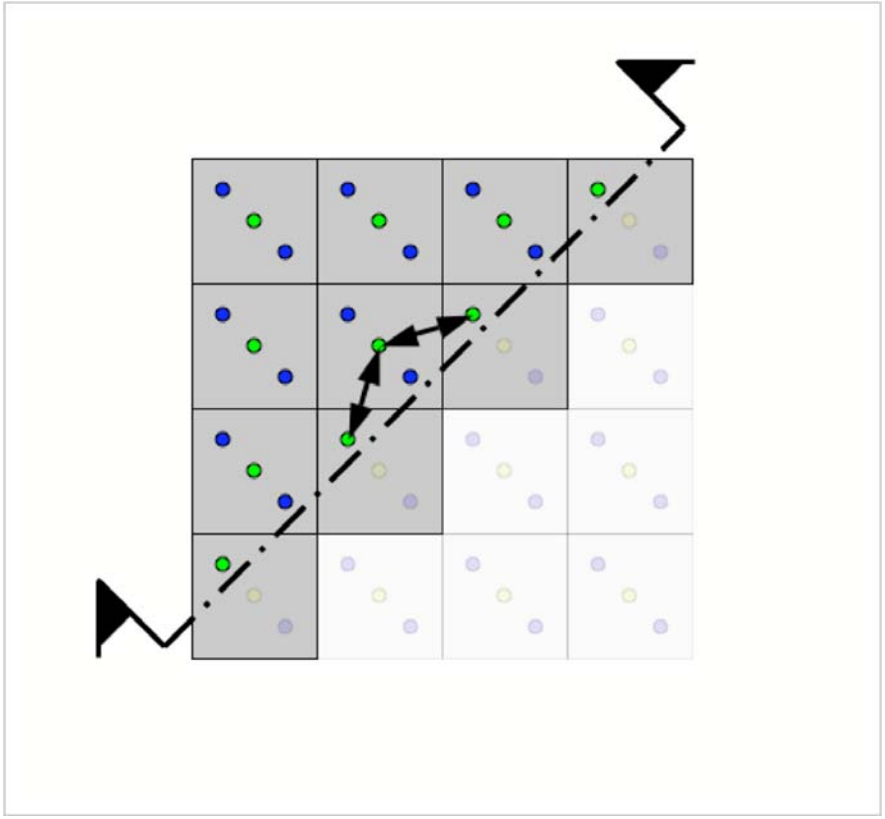
© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid



© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid

```
#version 120
centroid varying float myMixer;
// Interpolate color between blue and yellow.
// Let's do a sqrt for a funkier effect.
void main( void )
{
    const vec3 blue   = vec3( 0.0, 0.0, 1.0 );
    const vec3 yellow = vec3( 1.0, 1.0, 0.0 );

    float a = sqrt( myMixer );
    vec3 color = mix( blue, yellow, a ); // nonlerp
    gl_FragColor = vec4( color, 1.0 );
}
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid

```
varying float myMixer;
// Interpolate color between blue and yellow.
// Let's do a sqrt for a funkier effect.
void main( void )
{
    const vec3 blue   = vec3( 0.0, 0.0, 1.0 );
    const vec3 yellow = vec3( 1.0, 1.0, 0.0 );
    float mixer = max( 0.0, myMixer );
    float a = sqrt( mixer );
    vec3 color = mix( blue, yellow, a ); // nonlerp
    gl_FragColor = vec4( color, 1.0 );
}
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Centroid

```
#version 120
centroid varying float myMixer; // Beware of derivative!
varying float myCenterMixer;    // Safe for derivative.
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Invariant

- In OpenGL ES Shading Language
- Candidate for future GLSL

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Invariant

```
varying vec4 P;

uniform float scale, bias;

void main ( void )
{
    gl_Position = gl_Vertex * scale + bias;

}
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Invariant

```
varying vec4 P;

uniform float scale, bias;

void main ( void )
{
    gl_Position = gl_Vertex * scale + bias;
    P = gl_Vertex * scale;
}
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Invariant

```
#version TBD
varying vec4 P;
invariant gl_Position;
uniform float scale, bias;

void main ( void )
{
    gl_Position = gl_Vertex * scale + bias;

}
```

© Copyright Khronos Group, 2007 - Page

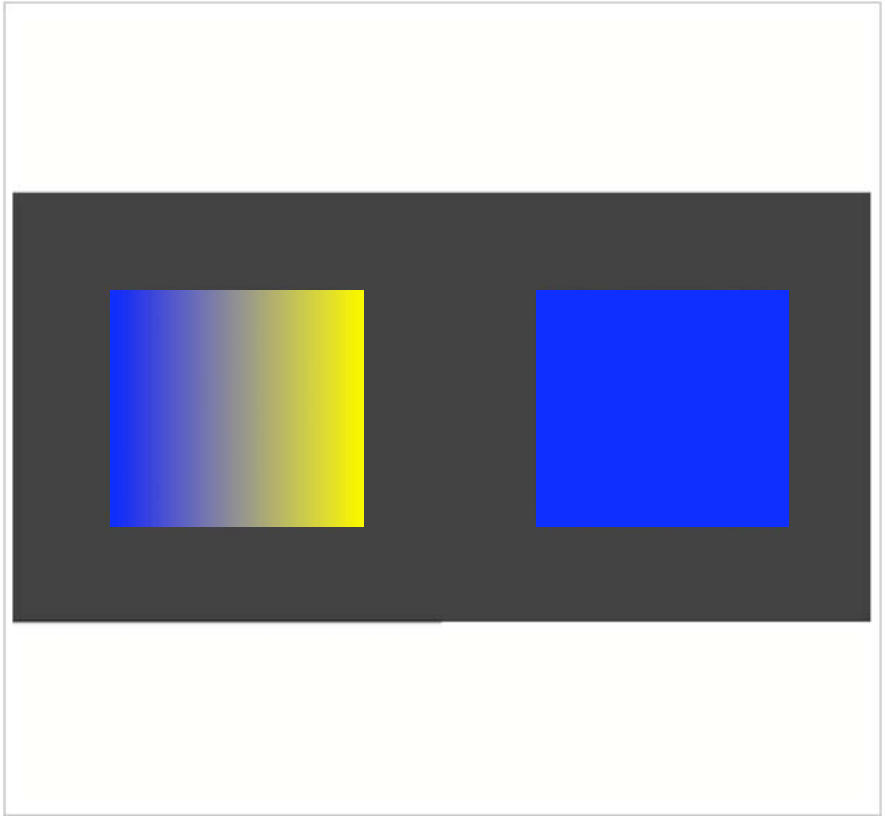
Cool GLSL Tips - Invariant

```
#version TBD
varying vec4 P;
invariant gl_Position;
uniform float scale, bias;

void main ( void )
{
    gl_Position = gl_Vertex * scale + bias;
    P = gl_Vertex * scale;
}
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Flat



© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Flat

```
#version TBD
flat varying vec3 myFacetNormal;

// fallback for prior versions:
// Hijack fixed function gl_Color,
// beware of possible precision issues.
// (Lookup games?)
//
// OR
//
// Break TRIANGLE_STRIP into TRIANGLES,
// beware of possible performance issues.
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Noperspective

- Some early GL implementations could be controlled with the hint:
  - GL_PERSPECTIVE_CORRECTION_HINT
    - NICEST
      - perspective correct interpolation
    - FASTEST
      - no perspective correct interpolation
      - for color, rarely for texture coordinates
    - DONT_CARE
      - Implementation choice
- Modern GL implementations ignore this hint
- Useful for some screen space algorithms

```
#version TBD
noperspective varying vec2 myLinearValues;
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Integer

- Current OpenGL Shading Language has limited support for int types
  - Floating point friendly!
    - Odd duck of int_least17_t
    - Several implementations are another odd duck int_least25_t
  - No bitwise operators
  - No unsigned integers
- Add PROPER signed and unsigned int
  - Equivalent to int32_t and uint32_t
  - Bitwise operators (~, &, ^, |, <<, >>)
  - Integer operator %
- Add integer varyings (must be flat shaded)
- Add integer samplers (no filtering)
- Add integer pixels (no blending)

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Integer

```
#version TBD
// New types
int i; // now int32_t
ivec2 i2;
ivec3 i3;
ivec4 i4;
unsigned int u; // new uint32_t
uvec2 u2;
uvec3 u3;
uvec4 u4;
// New samplers (not exhaustive)
isampler1D myShirtMap;
usampler2D myBootMap;
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Integer

- Example fallback - testBit

```
#if __VERSION__ TBD
#version TBD
bool testBit( const in unsigned int x, const in int bit )
{
   return ( x & (1 << bit) == 1 );
}
#else
// Test an 8 bit normalized texel as if int
bool testBit( const in float x, const in float bit )
{
   const float scale = 255.0;
   const float epsilon = 0.125/256.0;  //belt+suspenders
   float y = floor( x * scale + epsilon ); // x=[0.0,1.0]
                                           // y=[0.0,255.0]
   float mul = ( 0.5 / pow( 2.0, bit ) );
   float t = fract( y * mul );
   return ( t >= 0.5 );
}
#endif // __VERSION__
```

© Copyright Khronos Group, 2007 - Page

Cool GLSL Tips - Texture Arrays

- GLSL already permits array of samplers
  - BAD, very BAD, very very BAD

```
sampler2D mySamplerArray[4];
```

- Instead, consider new TextureArray

```
#version TBD
sampler2DArray mySamplerArray;
// Array “slice” is specified by texCoord.p
// No filtering between “slices
```

- For small number of slices, can fall back to array of samplers
  - BAD, very BAD, very very BAD

© Copyright Khronos Group, 2007 - Page

Cool GLSL tips - Questions?

© Copyright Khronos Group, 2007 - Page